AN ABSTRACT OF THE THESIS OF

Olivier Colle for the degree of Master of Science in Electrical and Computer Engineering presented on September 23, 1999. Title: Motion-Based MPEG Video Summarization.

Redacted for privacy

Abstract approved: 10-29-99

Wojtek J. Kolodziej

A new, fully automated summarization algorithm of MPEG compressed videos is designed to address the challenge of content based video retrieval. This algorithm segments the MPEG stream motion vector magnitudes using a seed growing region technique. Following the spatial segmentation of each frame, an intra-frame class-merging operation is performed. The resulting spatio-temporal classification is filtered to extract the video sequence main objects. The gravity centers of the identified objects are computed in background and fixed references, and are used to estimate the parameters of a 2-order ARMA model which summarizes the motion. Experimental results verify the effectiveness of the proposed algorithm and emphasize the usefulness of the MPEG motion vectors for object motion estimation.

Motion-Based MPEG Video Summarization

by

Olivier Colle

A Thesis Submitted
to
Oregon State University

In Partial Fulfillment of the requirements for the degree of

Master of Science

Presented September 23, 1999
Commencement June 2000

Master of Science thesis of Olivier Colle presented on September 23, 1999

Approved:

Redacted for privacy

10-29-99

Major Professor, representing Electrical and Computer Engineering

Redacted for privacy

Chair of Department of Electrical and Computer Engineering

Redacted for privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Olivier Colle, Author

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my major professor Dr. Wojtek Kolodziej for his guidance, advice and support with my research during my stay at Oregon State University. I appreciated his sense of humor, his challenging mathematical questions and his excellent editorial input during this research.

I thank Audrey for her trust, her love and her guidance during this work.

I wish to thank the people of the Electrical and Computer Engineering department for making my education possible especially the faculty, staff and fellow student I have worked with at the Modern Communication Center Laboratory.

I thank my family for its support and assistance, which were keys to the successful completion of this thesis.

TABLE OF CONTENTS

TABLE OF CONTENTS, CONTINUED

LIST OF FIGURES

LIST OF FIGURES, CONTINUED

Figure Page

LIST OF TABLES

To Audrey, Wistan and my family

MOTION-BASED MPEG VIDEO SUMMARIZATION

CHAPTER I : INTRODUCTION

I.1 Motivation

The rapid generation and distribution of digital information and in particular digital video requires development of new technologies for efficient classification and retrieval. One can now access selected clips from archives of thousands of hours of video footage almost instantly, yet the sheer volume of data makes any retrieval task overwhelming. Efficient browsing tools that allow user to retrieve a video sequence based on specific criteria are needed to maintain the video database systems.

The video archives such as the Hollywood movies or TV shows do not permit an effective manual classification. Sorting manually the existing digital media would require tremendous effort. Thus the need for new tools based on automated classification and automated retrieval is apparent. The main purpose of these new tools is to summarize a video sequence into a form that can be used to match a particular user query. Video summarization techniques can be roughly divided into two categories: key frame detection and content summarization.

The key frame technique attempts to detect significant information changes such as color, texture or motion, in several consecutive frames. If the change is large enough, the frame is retained under assumption that it summarizes the video content better than the other frames. Obviously, selecting one key frame may not adequately summarize the complex information of a frame sequence, which may include camera zoom and pan, light changes and object motion. Usually, in most current techniques, more than one key

frame is used to represent the footage. Still, one of the main problems of this summarization technique is that it does not consider small, yet important changes between key frames. For example, a ball that appears on the left side of the screen and bounces to the right side will produce generally only one key frame. Therefore the ball motion cannot be tracked during this short sequence. While facilitating the retrieval process, most of the existing tools allow only to browse the video key frames without accepting any filtering criterion from the user.

In content summarization, attributes such as shape, texture, color and motion are identified and summarized along the video sequence. According to specific criteria, the user can retrieve all videos matching a particular request. This summarization technique is more precise and specific than the key frame approach. The user can specify color, position, motion, duration, etc. Consider for example the search for all sequences containing a black, round falling object. With a key frame-based-search engine, such a request is strenuous and the user will have to browse many video keyframes before finding one that matches these criteria. However in content summarization, few trials will identify the desired video sequences. Content summarization techniques are usually more complex since a field of information needs to be created. Also a metric indicating how well the user request matches the video has to be implemented. Among new technologies of content based search engines, few use the concept of object motion and almost none of them work directly in the compressed domain.

In this thesis, we propose to design a new, fully automated summarization algorithm of MPEG compressed videos. Since motion is an important and useful cue in human perception, our summarization algorithm is primarily concerned with motion.

Other attributes such as shape or color might be added to the summarization features in future work. Also, in order to obtain an efficient implementation, only compressed information provided by the MPEG stream is used.

The developed algorithm could be part of a complete retrieval system. After specifying motion criteria, the user searches for video footage on the Internet using a basic Internet browser. In contrast to key frame methods, the browsing of the retrieved video sequences is restricted only to those matching the motion criterion.

The complete system would perform two different actions: tagging and retrieving. The tagging process starts with a MPEG compressed video file located on a local or remote computer. After opening the video file, the algorithm identifies moving objects, summarizes the motion and appends it to the MPEG file header for subsequent retrieval uses. During the retrieval process, the user stipulates motion attributes such as position and velocity. The algorithm searches for matching video sequences in the local computer and in any remote video database specified by the user. While searching, only the header of the video file is opened and compared to the request resulting in an efficient search.

The developed here algorithm performs object motion identification and summarization which should be tagged to the MPEG files. Development of the complete retrieval system is beyond scope of this work and it is suggested for further work.

I.2 Background

I.2.1 MPEG video compression standard overview

There exist many video compression methods. However, the MPEG standard created by the Motion Picture Expert Group, is the most widely used. Since the first

MPEG-1 standard, created in 1992, many MPEG standards have been released for video as well as for audio : MPEG-2, MPEG-3, MPEG-4, and, currently under development, MPEG-7. Some key points common to all MPEG standards include:

- MPEG is a group of people that meet under International Organization for Standardization (ISO): ISO 11172 for MPEG-1 (5 parts), ISO 13818 for MPEG-2 (9 parts), ISO 15938 for MPEG-7.
- MPEG is application specific, as it does not attempt to solve all application needs but rather offers syntax designed to compromise between cost and quality or equivalently, between computational complexity and compression ratio.
- The computationally intensive work is done on the encoder side permitting systems with small computational resources to decode MPEG stream.

Currently, MPEG-2 is widely used for high quality video, including DVD, HDTV and Satellite Broadcasting while MPEG-1 is found in applications that do not require more than Video Home System (VHS) quality video or that use a limited bandwidth. Being Internet oriented (limited bandwidth), our algorithm works primarily with MPEG-1 compressed sequences. The developed here technique can easily be extended to MPEG-2 video sequences since the differences between the two standards do not affect significantly the proposed segmentation scheme.

The MPEG-1 video compression standard exploits both the spatial redundancy within each frame of a video sequence (the high likelihood that neighboring pixels will be similar) and the temporal redundancy between successive frames (the high likelihood that

neighboring frames will be similar). The following discussion briefly describes how these redundancies are exploited.

The key points of MPEG-1 are:

- Block-based video scheme.
- Local decorrelations are based on hybrid Discrete Cosine Transform – Quantization – Variable Length Coding (DCT-Q-VLC).
- Dead-zone quantizer.
- Displaced Frame Difference (DFD) uses a quantized prediction error.

As shown in Figure I-1, the input of a MPEG-1 encoder is typically an ordered sequence of frames, each frame being represented by a set of 24 bit RGB pixels. The RGB values are first converted to 12 bit YCbCr format. This conversion disassociates the luminance from chrominance components to obtain a video stream format similar to television color system (YPrPb). Human visual perception is also more sensitive to luminance than chrominance. By allocating 8 bits for the luminance component Y and 2 bits each for the chrominance components Cb and Cr, this color conversion emphasizes the signal luminance and produces a compression ratio of 1:2. Each frame is then broken into 16x16 block of pixels, known as macroblocks, which are then split into blocks assigned exclusively to Y, Cb or Cr. For example, in the 4:2:2 MPEG-1 standard, the Y component is divided into four 8x8 pixel blocks and the Cb and Cr components are divided into two 16 width by 8 height pixel blocks.

Each frame and each macroblock is assigned a type, according to a user-defined type pattern. There are three types of frames: I-frames, P-frames and B-frames.

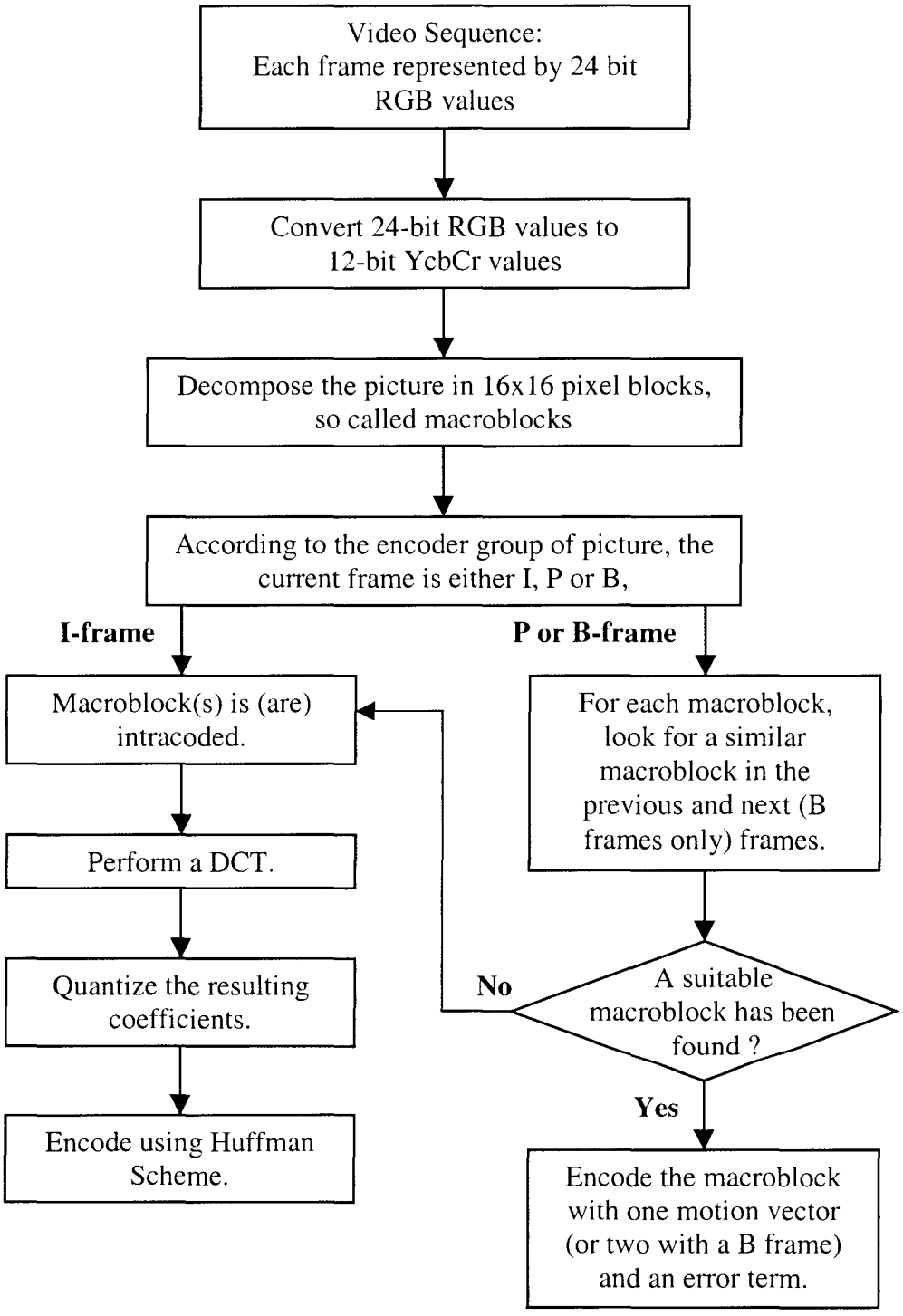


Figure I-1: Scheme of a MPEG-1 encoder

I.2.2 Different MPEG frame types

In an Intracoded (I) frame, a Discrete Cosine Transformation (DCT) is applied to each block to convert them from the spatial to the frequency domain, concentrating the energy (i.e. the most significant information) at the top left corner of the block. The resulting DC and AC coefficients are then quantized according to a user-defined "q-matrix" and "q-scale". Each coefficient is first divided by the product of the corresponding entry in the q-matrix and the q-scale and then rounded down to the nearest integer. This is a lossy step in the compression, and gives the user a mechanism for trading quality for bit rate by modifying the q-scale. DCT and quantization together exploit spatial locality: when colors do not change much from pixel to pixel, the frequency is lower and more regular across the block. It can therefore be represented with relatively few low-order DCT coefficients. This technique is closely related to the still image compression represented by the JPEG standard.

In a Predicted (P) frame, when possible, macroblocks are encoded relative to the previous "reference" frame, which must be another I or P-frame. The encoder, using a search algorithm, looks for macroblocks in the reference frame that are similar to the given macroblock. The MPEG standard does not specify the search algorithm whose complexity usually depends on the available resources in the encoder. If a suitable macroblock is found, the block is encoded as a motion vector (i.e. the offset from the macroblock being encoded to the reference macroblock) and an error term. A motion vector has two integer components that represent the displacement in units of whole or half pixels. If no suitable reference macroblock can be found, the macroblock is intra-coded (i.e. encoded as in an I-frame).

Finally, a bidirectional (B) frame is encoded relative to the previous reference frame and/or the future reference frame. The previous reference frame is the last I or P-frame before the B-frame, and the future reference frame is the next I or P-frame following the B-frame. The encoding of macroblocks in B-frames is similar to the encoding in P-frames, except that the macroblock might be represented by two motion vectors (i.e. the macroblocks are averaged): vector to the future macroblock and vector to the previous macroblock, or simply by the DCT coding. Figure I-2 shows how a series of I, P and B-frames is encoded relative to each other.

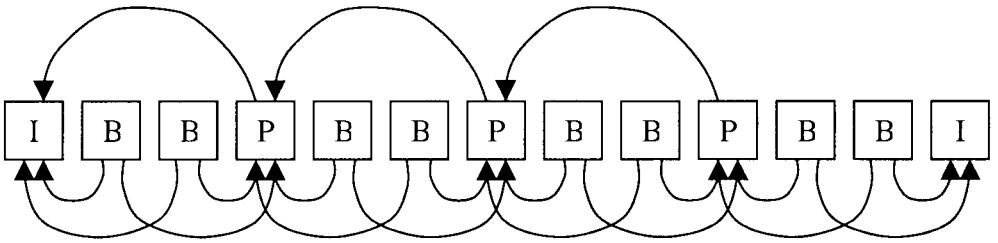


Figure I-2: Frame dependencies in a group of picture

P and B-frames exploit temporal locality exhibited by most video streams. If each frame in a sequence was similar to its neighbors, one expects the P and B-frames to be significantly smaller than the I-frames since the macroblocks have a close match in neighboring reference frame. From this brief description of MPEG-1, we observe that performance of our algorithm depends on the encoder as follows:

- The search pattern for reference macroblocks, that is, how the best reference macroblock that yields the smallest error term is found and how large is the search region.
- The frequency of P-frames in the group of pictures.

I.3 Notation

The following notation is defined:

- Let F be the set of N P-frames and f_p be the current frame being processed by the algorithm, where $f_p \in F$ and $1 \leq p \leq N$. N represents the number of P-frames of the entire video sequence. S_X and S_Y are respectively the horizontal and vertical frame sizes measured in macroblocks.
- Each frame f_p is organized as a matrix M_p of macroblocks. Elements of M_p are denoted by $m_p(x, y)$, $1 \leq x \leq S_X$, $1 \leq x \leq S_Y$. Similarly to the MPEG scheme, the origin is chosen to be the top left corner and the corresponding macroblock is $m_p(1, 1)$.
- Each macroblock $m_p(x, y)$ is a triple: ($Vx_p(x, y)$, $Vy_p(x, y)$, $r_p(x, y)$), where $Vx_p(x, y)$, $Vy_p(x, y)$ are the two components of the motion vector $V_p(x, y)$ and $r_p(x, y)$ is the identification number of the region to which the macroblock belongs to.
- Let T be the set of all regions resulting from the spatial segmentation and $N_R(p)$ the number of regions for the p^{th} frame.
- Let $\rho_{i,p}$ be the i^{th} region of the P-frame p (with $1 \leq i \leq N_R(p)$). A region $\rho_{i,p}$ is composed of similar macroblocks resulting from the spatial segmentation:

$$\rho_{i.p} = \{m_p(x, y) \in M_p, r_p(x, y) = i\} \qquad \text{(Eq I-1)}$$

I.4 Overview of the thesis

The objective of this research is to build an efficient motion-based MPEG video summarization algorithm using the information provided by the motion vectors from the

MPEG stream. After precisely defining the meaning of motion vector, a suitable algorithm that segments the motion vectors in both space and time is defined. Following the segmentation, noise is filtered and objects with consistent motion as well as the background are identified. Finally, object motion and background velocity are extracted and summarized using an appropriate model.

The thesis organization is as follows. Chapter II presents a review of previous relevant works on spatio-temporal segmentation, growing region algorithm and motion-based video retrieval systems. Chapter III presents the methods used to segment the predictive motion vectors, to identify the main objects and to find a suitable model summarizing motion information. In Chapter IV, the simulation results are presented and the summarization efficiency is discussed. Finally, Chapter V presents conclusions and recommendations for the future work.

CHAPTER II : LITERATURE REVIEW

This chapter reviews the current research in content-based video summarization and retrieval. As they constitute the backbone of retrieval systems, spatio-temporal segmentation algorithms are first presented. Next, growing region algorithms are discussed and some inherent problems are emphasized. The chapter concludes by reviewing some content-based video retrieval systems.

II.1 Previous work in spatio-temporal segmentation

Spatio-temporal segmentation techniques attempt to identify the objects present in a scene, based on spatial and temporal information [1], [2]. Spatio-temporal segmentation plays a fundamental role in computer-assisted scene analysis such as video retrieval. It forms the backbone of schemes for recognizing, classifying and tracking objects. In [3] spatial information is defined as the brightness attributes of the video sequence. Using MPEG compressed information, brightness might be replaced by the DC component, which represents approximately, the brightness of the macroblock. The motion vectors, computed using the color information, also include a spatial information: consider for example a non-deformable object moving against a background. Assuming that there are no camera special effects, all points of this object move at the same speed and the motion vectors for each P-macroblock of this object will approximately have the same components. Depending on the encoder macroblock search algorithm we might loose track of some parts of the object but obtain still an acceptable spatial segmentation. Temporal information is defined by the motion information of an object. According to the

Gestalt "law of common fate" [24], meaningful regions are retained if they are defined on the basis of temporal coherence.

Techniques for spatio-temporal segmentation can be generally grouped into two classes. The first class uses a top-down approach as they sequentially extract the different objects forming the scene by estimating the parameters of dominant motions. Regions complying with the current dominant motion are assumed to belong to the same object and are not considered in the next iteration.

In contrast, the second class has a bottom-up approach. These typically start with an oversegmentation of the video sequence and iteratively merge regions in order to obtain a coherent and stable scene description [3], [4]. Different proposals have been made for spatio-temporal criteria used to merge regions together. In [4] a spatio-temporal Similarity Measure is defined, which exploits both spatial and temporal information while placing more emphasis on the temporal part. Two statistical tests, for temporal and spatial similarity, are used in a weighted combination to merge two candidates. A modified Kolomogoroff-Smirnov test determines the level of coherence between the motions of two regions. Spatial merging is accomplished using a statistical test, which determines the probability that two regions share spatial characteristics. [6] proposes a spatial-segmentation method based on obtaining a minimum description of motion between two consecutive frames in a long sequence.

II.2 Region growing algorithm

Current research on growing region algorithms focuses on pixel intensity segmentation and only few are designed to segment motion information. In this thesis

these algorithms are generalized and pixel intensity is replaced by the macroblock motion vector magnitude.

Growing-merging region algorithms consider simultaneously the various information, here, the motion vector magnitude and its spatial distribution. Thus, the resulting classification is usually composed of spatially compact regions having fairly concentrated histograms. Applying the growing-region requires the selection of criteria to append an element to an existing neighboring class. Growing region algorithms can be divided in two categories: single and global linkage.

II.2.1 Single linkage algorithms

One of the simplest region growing methods is by single linkage. Two elements, such as macroblocks or pixels are joined together if they have *similar properties*. For example, two macroblocks from the same frame are considered related if the magnitude of their motion vectors is smaller than a fixed threshold.

Although attractive due to its simplicity, this technique requires a good knowledge of the motion vector distribution.

[21] introduces an improvement in single linkage growing algorithm by replacing the fixed threshold with a function of motion vector magnitude in the macroblock neighborhood.

Using this method improves the segmentation but a undesirable chain effect may occur. Although attractive in their simplicity, single linkage growing region algorithms have the disadvantage of considering only the local information.

II.2.2 Global linkage algorithms

In global linkage region growing algorithms, macroblock linkages are performed by comparing the current macroblock motion vector magnitude with existing neighboring region statistics. In [9] it is suggested that a macroblock is appended to the neighboring region if its motion vector magnitude is close to the region average motion vector.

Similarly, [7] uses the standard deviation as the criterion and a macroblock is merged into one of its neighboring region if the resulting region variance stays lower than a fixed threshold. While interesting since comparisons are performed on a set of macroblocks, this method has its disadvantages. Large classes may absorb non-consistent elements and still maintain a low standard deviation. Also during the algorithm initialization, the regions are composed of few macroblocks only and variance is statistically meaningless. To solve this problem the growing process can be performed through morphological dilation operations. As shown in Figure II-1, instead of appending a single macroblock, a macroblock set is appended, and meaningful statistical comparisons are obtained faster.

II.2.3 Initial seed

Both of the above algorithms require as input a single or multiple initial seeds. This is one of the growing region algorithm weaknesses. In medical applications, where the segmentation is performed on few images, a user can specify the initial seed [7] [9]. For example, a user can click on different image parts providing a priori classification.

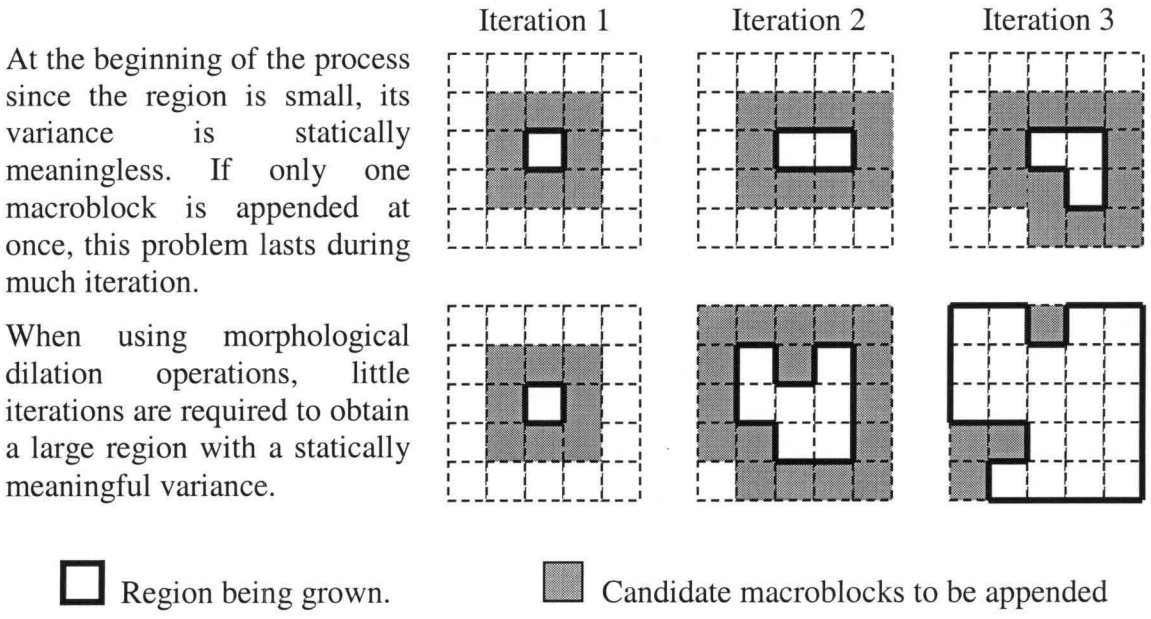


Figure II-1: Morphological dilation operations

In video database applications, segmentation is performed on large set of images and automatic seed computation is required. Similarly, the number of regions is usually unknown and must be initially chosen by the user. Thus, most of the existing algorithms require human inputs.

II.3 Content-based video retrieval systems

A large number of catalogs and search engines indexes the plethora of documents on the World-Wide Web. Some systems, such as Yahoo, Lycos and Alta Vista, index the documents by their textual content. These systems periodically scour the Web, record the text on each page and through automated analysis, condense information into compact and searchable indexes. However few video-searching tools are currently available. New videos are added and removed everyday on the World-Wide Web yet no efficient systems

exist to search and retrieve them. This sub-section describes latest research on content-based video retrieval systems designed for the World-Wide Web.

Content-based technologies have enabled recent advances in management and search of visual information. User can retrieve images and videos using salient visual features such as color, texture, shape, motion and spatial information. In [11] an interactive indexing tool called FINDIT is developed. This system allows to locate images from a large image set. Images are classified based on their normalized histogram content and a retrieval algorithm compares the query with the indexes.

More recently, tools based on content-based visual search engine called WebSEEk and MetaSEEk were developed [10]. The main features of these systems are:

- Queries are based on the visual features of the data,
- Scenes are automatically grouped into visually homogeneous clusters,
- Browsing and navigation is done by content through an image and video archive.

After collecting videos from the World-Wide web, the system indexes them using key words and color histograms. A visual query allows the user to specify a color histogram distribution and a video category name. The retrieval algorithm searches for histogram similarities in the specified category and displays all matches. A relevance feedback is used to reformulate the query and obtain successively a better retrieval accuracy. Although this system is currently used, it requires user input to pre classify the videos and images. The content-based engine is also limited to the color histogram information.

Another approach is to retrieve video by still image analysis. ImageMiner™ [12] developed at the University of Bremen uses mainly two techniques for video retrieval: shot detection and *mosaicing*-technique. To reduce the large amount of video frames, a primary histogram-based shot detection algorithm is performed. Next the amount of frames is reduced by selecting key-frames of a shot.

In [12] the problem of key-frame selection is discussed. Choosing the first and the last frame is not always suitable thus the *mosaicing*-technique is introduced to condense the amount of video data without loosing as much information. The basic idea of this technique is the creation of a single image for each shot which summarizes the complete information of the scene. Although the *mosaicing*-technique provides good result on landscape footages, it requires a high computational time. The algorithm is also not able to handle object movements which do not match the movements of the whole image.

[13] presents a content-based MPEG video retrieval system that uses example video clips as a query. The algorithm primarily analyses each MPEG video sequence and derives its signature. This signature is constructed from the DC coefficients and the motion information. As shown in Figure II-2, the DC signature is extracted using N window pairs in each frame defined by the user. The DC coefficients of each pair are compared and if their difference is larger than a fixed threshold the bit signature S_i is set to 0, otherwise it is set to 1. N influences directly the signature bit length. For example, for a video sequence containing only I-frames, a 128 bit signature is derived from each I-frame, where 64 bits are derived from the luminance component and 32 bits from the two chrominance components with the 32 bits reserved for motion set to zero.

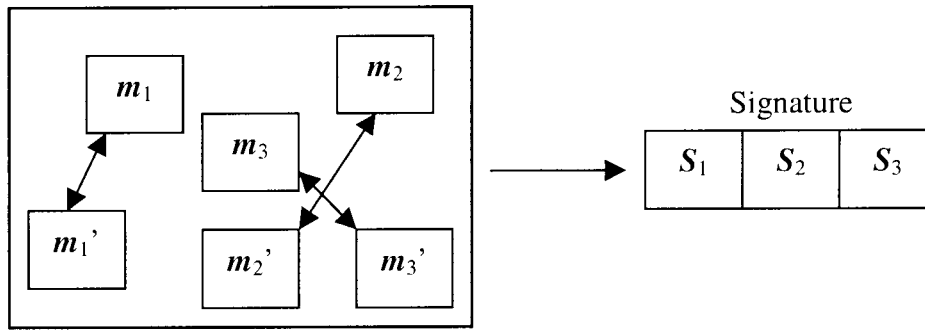


Figure II-2: 3 window pairs are chosen to compute the signature

The motion signature computation requires a basic 4-entry table. A 00 value is generated if no motion is present, a 01 if the first window has motion information, 10 if the second has motion information and 11 if both have motion attributes.

The query is performed by providing an example video clip. The corresponding signature is extracted and compared to the existing video clip signatures. Although time efficient, this method does not always perform suitable retrievals. In many cases, the user must modify the default window location to a more specific position in order to improve the retrieval efficiency.

Content-based video retrieval is in its early stage. Only few systems use content attributes and none of them provide an efficient retrieval. While focusing on color content, most systems still require a human interaction to a priori classify the media.

CHAPTER III : SPATIO-TEMPORAL SEGMENTATION

The spatio-temporal segmentation is the heart of our video summarization system. In order to retrieve video sequences according to object motion attributes, the segmentation algorithm needs to distinguish a moving object from others and from a fixed or moving background. While filtering noise it has to retain small or large objects with coherent motion. Finally these criteria must be satisfied in presence of such camera effects as zooming and panning.

The developed segmentation algorithm takes as input the forward motion vectors magnitude from the P-frames of the MPEG stream and segments them to obtain consistent object motion. As shown in Figure III-1, the segmentation algorithm proceeds in three steps. In the first step, the field of the forward motion vector magnitude is segmented using an advanced growing region algorithm. The small regions are then merged with the largest regions to obtain the most significant classes in the current frame. Region "time" merging is performed by relating the regions between frames according to their predicted position and real position. All of these steps are performed for each P-frame of the video sequence until the end is reached. The input and output of the algorithm and each of its steps are discussed in the next sections.

III.1 Segmentation input

The spatio-temporal segmentation technique proposed in this work identifies object motion using only the forward motion vectors from the MPEG stream. The compact size and easy accessibility of this information increase the segmentation speed, permitting near real-time processing.

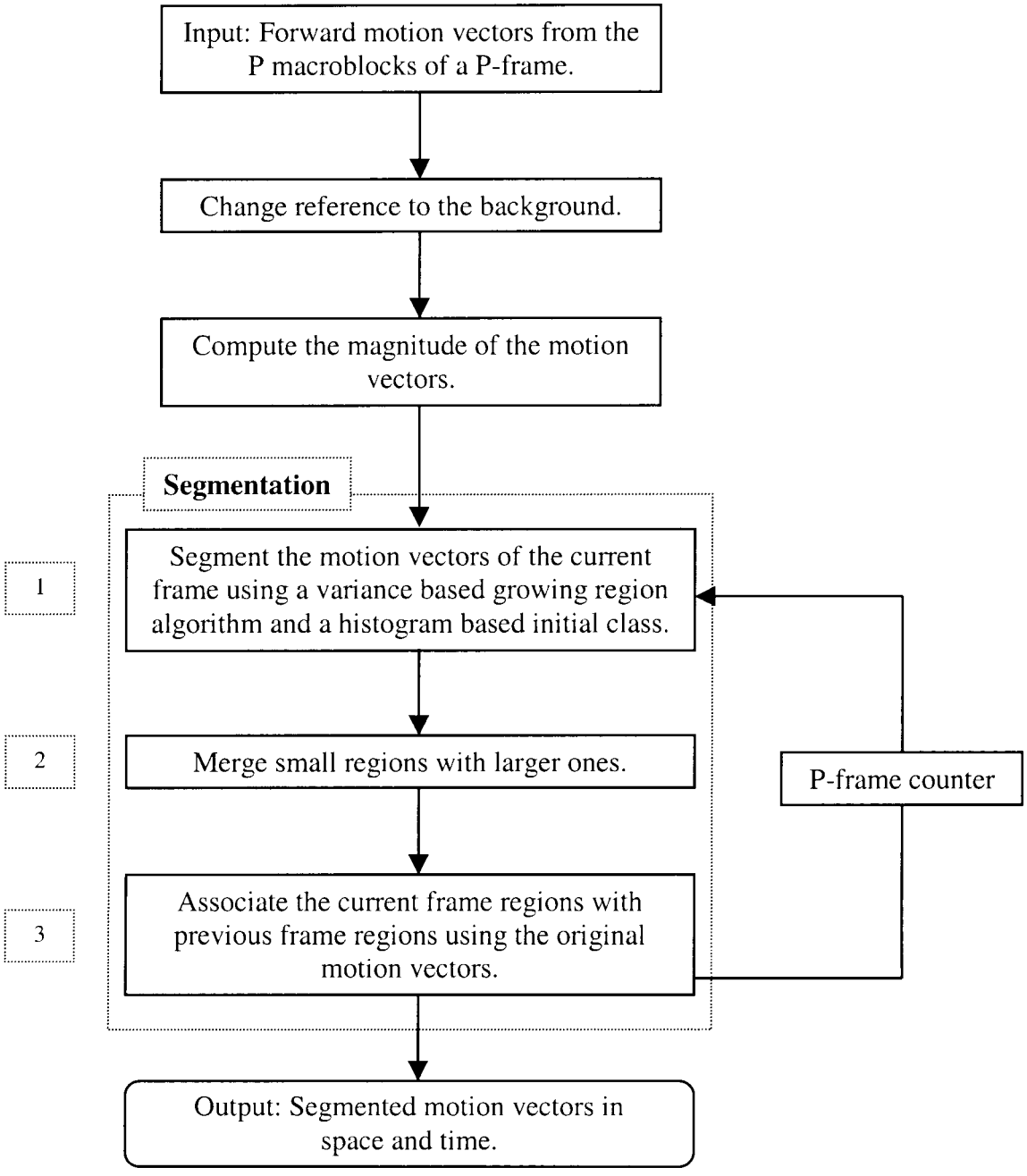


Figure III-1: Segmentation algorithm

III.1.1 Motion vectors from P-frames

In MPEG-1 stream, motion vectors reside only in the P and B-frames. Thus, I-frames, which do not provide any direct motion information, are not used by the segmentation algorithm. B-frames contain both backward and forward predictive motion vectors. The backward component allows to increase the compression ratio however B-block computation increases considerably the complexity of the MPEG encoder. Similarly, the complexity of the segmentation algorithm increases, however without any significant improvement of the summarization performance. Most of MPEG-1 encoders use a group of pictures where two B-frames separate two P-frames. Thus, in the NTSC standard which has video sequence rate of 30 frames per second, P-frames appear every 3/30sec or every 6/30sec if surrounded by an I-frame. Motion vectors are available in the worst case every 0.2s and seldom does an object significantly change its trajectory in this short period of time. Accordingly object motion exceeding 5Hz frequency may become undetectable.

Here, only the forward motion vectors coming from the P-frames are utilized in the segmentation algorithm. These P-macroblocks as well as their motion vectors are encoder dependent. In a P-frame, depending on the macroblock search algorithm, a macroblock will be encoded as an I or a P-macroblock, reducing or increasing the number of motion vectors and accordingly the motion information present in the frame. Most encoding algorithms search for a reference macroblock in a range of 63 pixels around the current macroblock. However some encoders decrease their processing time by decreasing the search area size to 31 or 15 pixels. Thus better macroblock search algorithm in the encoder gives higher segmentation accuracy.

III.1.2 Reference of the motion vectors

Video sequences with moving background are common, creating an illusion that moving object is stationary. This usually occurs if the camera is moving (e.g. placed in a car or a plane), or if zooming or panning effects are used. An example of the effect of panning camera on the field of the motion vectors is shown in Figure III-2. Except some frame border effects, all motion vectors have the same horizontal direction and the same magnitude, characteristic to the panning camera movement. A constant camera motion does not affect the segmentation, but in the presence of significant acceleration, region matching between the frames becomes harder. An object may appear to change its speed discontinuously while moving at the same speed or accelerating smoothly. To avoid these effects and obtain a better segmentation, the background reference must be selected prior to the segmentation.

To identify the background, we make an assumption that most of the belong to the background. Thus, most of the motion vectors represent the background motion. This conjecture does not remain valid when the object is much bigger than the background (viewed as the fraction of a frame). In this case, the reference becomes the object itself and is still suitable for the segmentation purposes. The worst case will happen when the object has the same size as the background. In such a case, we could propose an iterative process that, after initial segmentation, analyzes the sizes of the background and objects and, possibly, performs a new segmentation without changing the reference.

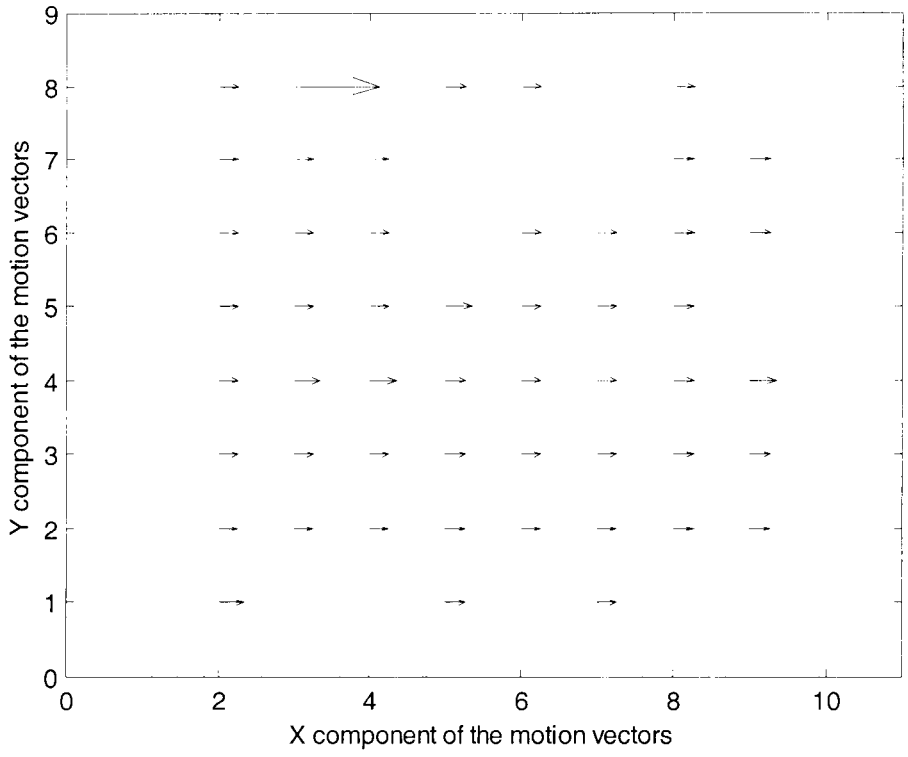


Figure III-2: Motion vectors during a panning camera movement

Changing of reference consist of subtracting in each frame the motion vector mean from the individual motion vectors. Given any frame f_p of F, the corrected forward motion vectors V_p are computed using the following:

$$\forall f_p \in F,\ \forall V_p \in f_p,\ W_p = V_p - \overline{V_p}\ ,\ \text{with}\ \overline{V_p} = \text{mean}_{V_p \in f_p}(V_p) \qquad \text{(Eq III-1)}$$

The above ensures in most cases that the motion attributes become camera or background motion independent.

III.1.3 Magnitude of the motion vectors

A forward motion vector holds two components, predicted vertical and horizontal displacement, which represent the position of the reference macroblock relative to its

previous position. Thus, as shown in Figure III-3, it provides a backward motion information.

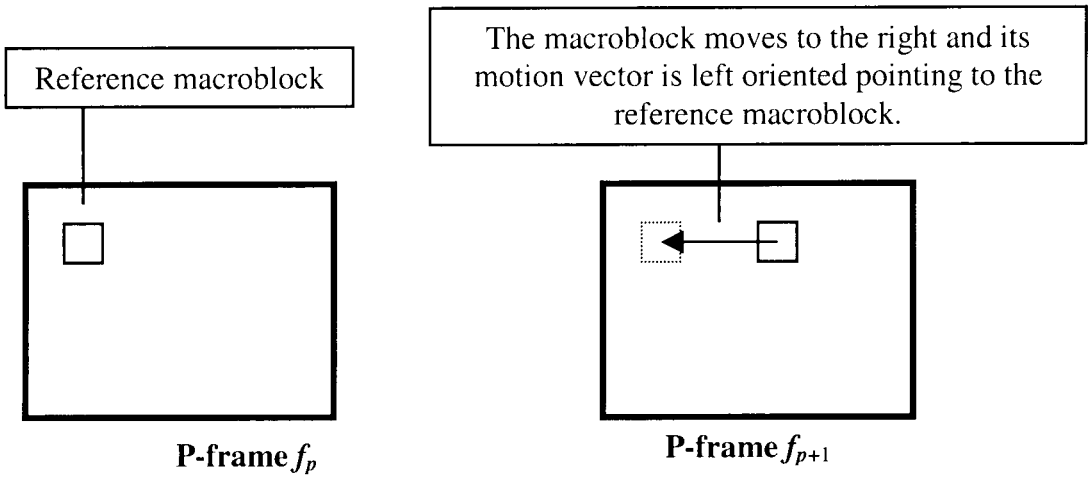


Figure III-3: Forward motion vectors provide reverse motion information

Two different approaches can be used to segment the motion vector set. The first one combines the X and Y motion vector components in order to perform directly one classification. The second method attempts to segment separately the X and Y components and next, groups the two resulting segmentations to produce only one classification. While loosing a part of the information, the first technique, which performs only one segmentation, is faster. The proposed segmentation uses this method and has the motion vector magnitude as the input. This approach however faces the following challenge. Considering two neighboring objects moving in opposite directions at the same speed (see Figure III-4). In the first frame after they start moving, using only the magnitude of the motion vector, it's very difficult to distinguish the two objects. However due to the opposite direction motion, there is a frame where the two objects will

be disjoint and where the spatial segmentation will distinguish the two objects from each others.

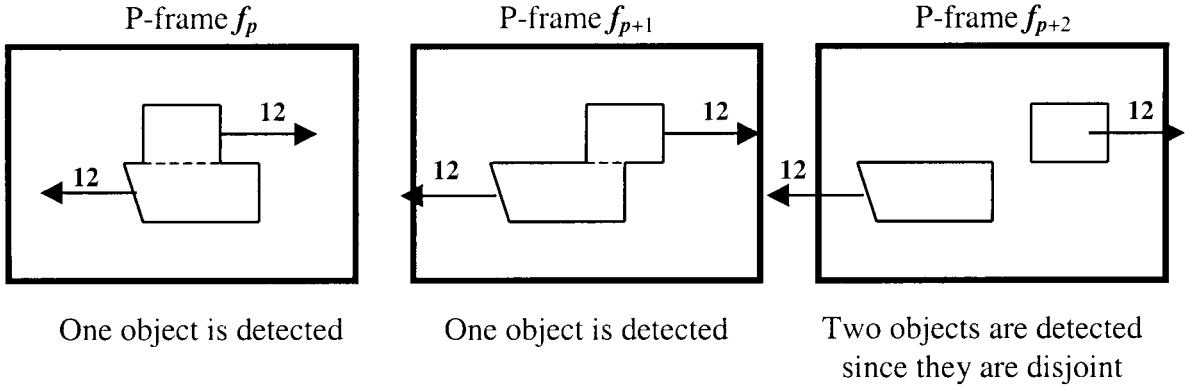


Figure III-4: Neighboring objects moving at the same speed in opposite directions.

Therefore, using the magnitude of the motion vector generally allows to identify the objects while maintaining a fast segmentation. The segmentation input being defined, the spatio-temporal algorithm can be described.

III.2 Spatio-temporal segmentation

This section describes the proposed spatio-temporal segmentation algorithm. The segmentation is the backbone of the system since it classifies the motion vectors to obtain the frame sequence main objects. The system relies on this segmentation algorithm and its robustness with respect to the encoding scheme. As discussed in I.2.1, the block-matching algorithm is encoder dependent and it is not designed for retrieval applications. Thus, the motion vectors are not fully suitable for segmentation. However, by considering the following points in the algorithm design, a proper classification can be obtained:

- In the case of a uniform color, moving object, the motion vector magnitude is reduced to zero. As shown in Figure III-5, consider two neighboring macroblocks of the same color. In frame f_{p+1}, one of the blocks occupies the position of another other block in the frame f_p. The color of this second block remains the same in both frames f_p and f_{p+1}, setting the corresponding motion vector to the null vector.

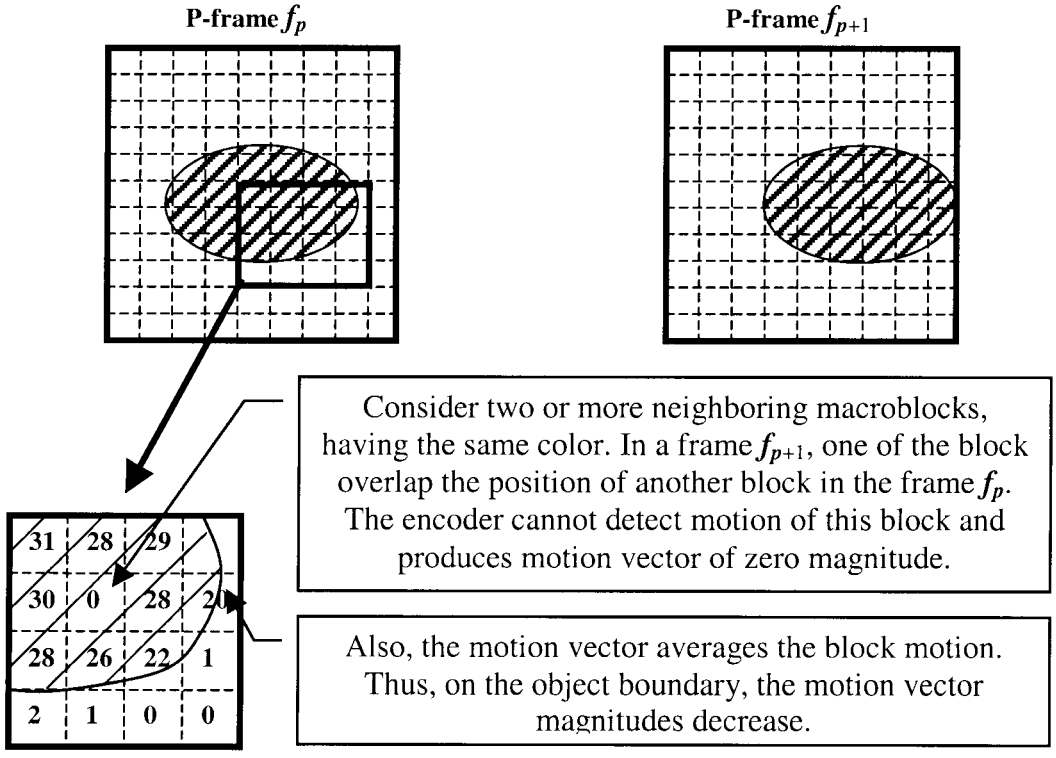


Figure III-5: Motion vectors estimation weaknesses

- The motion vectors allow to distinguish easily between a stationary and moving object but may encounter difficulty with segmentation of two neighboring objects. As shown in Figure III-5, the object boundary macroblocks usually belong to both the object and the background. Since the

motion vectors of this 16x16-pixel block represent the motion average the calculated value is incorrect in this case. Thus, the motion vector distribution of an object is usually wide, making their grouping more difficult. As shown in Figure III-6, this is especially important for objects with a high boundary-surface ratio. Their motion vectors are concentrated on one of the histogram sides masking the real object motion.

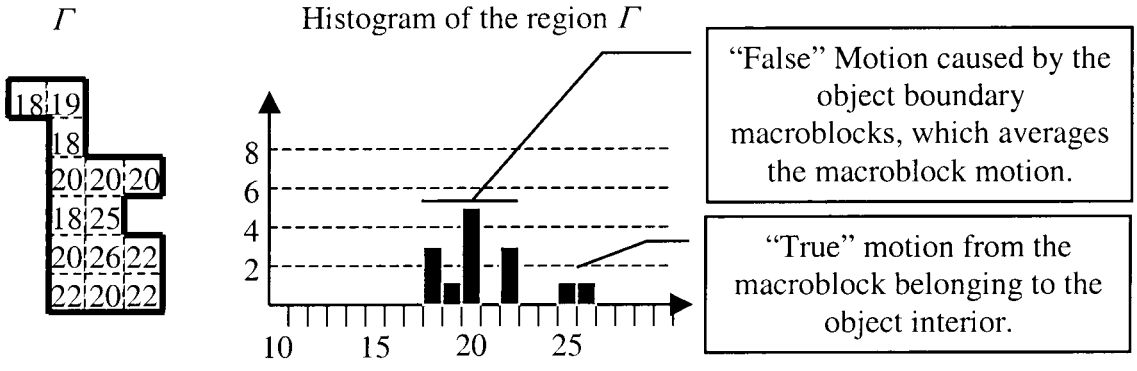


Figure III-6: Motion vectors histogram for a high boundary-surface ratio object

- In the encoder, the macroblock search algorithm may fail to find a suitable reference. Therefore, the macroblock is intra-coded and the corresponding motion vector is lost. This situation may occur when the light conditions change rapidly or when an object moves too fast to stay in the macroblock search algorithm range. However modern encoders use a large 64-pixel-range search area which alleviate this problem.

Taking into account the above, the spatio-temporal segmentation algorithm is designed as follows :

1. Segment the current frame motion vector using a variance based region growing algorithm and a histogram based initial seed.
2. Merge small regions with bigger ones.
3. Associate current frame regions with the regions of previous frames.
4. Proceed to the next frame until reaching the last one.
5. Filter the resulting classification keeping only "time persistent" objects.

III.2.1 Variance based spatial growing region algorithm

Chapter II discussed various still image segmentation methods. Growing-merging region algorithms consider simultaneously the main information, in our system the motion vector magnitude, and its spatial distribution. Thus, the resulting classification is usually composed of spatially compact regions having concentrated histogram. Applying the growing-region requires the selection of criteria to append an element to an existing neighboring class. From the existing solutions to this problem, we choose a variance-based criterion, which has distribution adaptive features and hence performs an efficient segmentation. Another challenge is in the initial seed choice. While many algorithms require a human input, the proposed algorithm must stay fully automated.

The algorithm combines spatial neighborhood and histogram neighborhood information through suitable morphological operations [7]. It retains the concept of reconsidering the macroblock class membership but changes the criterion of macroblock merging to avoid "large class absorbing effects". In contrast with [7], no assumption on the class number is required to perform the segmentation and no human interaction is

requested to obtain the initial seed. Thus, the segmentation algorithm is completely automated.

III.2.1.1 Initial seed

The first task is to identify either the background or the largest object as represented by the current frame dominant motion. The initial seed must be suitably chosen so that after its expansion, the resulting class represents the background or the largest object.

As shown in Figure III-7, a 10-bin (number of bins arbitrarily chosen) histogram of the motion vector magnitude of Λ_p is first computed, where Λ_p is the set of all motion vectors without membership (hence, in the first iteration, $\Lambda_p = \boldsymbol{M}_p$). The centroid η_p of the largest histogram bin is computed, representing the background (or the largest object) average motion.

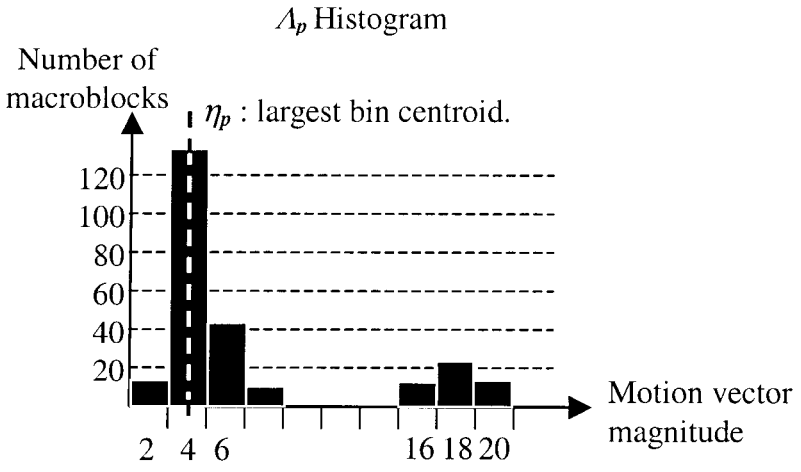


Figure III-7: Histogram computation used to obtain the initial seed

Next, η_p is used to form Ω_p, the set of all motion vectors with a magnitude the closest to η_p:

$$\Omega_p = \left\{ m_p(x,y) \in M_p, \left| \left\| V_p(x,y) \right\| - \eta_p \right| \leq \min_{m_p(x,y) \in M_p} \left(\left| \left\| V_p(x,y) \right\| - \eta_p \right| \right) \right\} \qquad \text{(Eq III-2)}$$

Thus, all motion vectors from Ω_p are suitable candidates to compose the initial seed since their magnitude is close to the dominant motion. However, instead of choosing randomly a member of Ω_p, which may be spatially isolated, the initial seed Γ is a closest macroblock to the spatial centroid m_c of Ω_p:

$$\Gamma = \{ m_0 \}$$

$$m_0 \in \left\{ m_p(x,y) \in \Omega_p : \left\| m_p(x,y) - m_c \right\| \leq \min_{\tilde{n} \in MC} \left(\left\| m_p(x,y) - m_c \right\| \right) \right\} \qquad \text{(Eq III-3)}$$

As discussed in the next sub-section, the initial seed Γ is enlarged through a growing-reducing algorithm to identify the background or the largest object.

Afterwards, Λ_p is reduced to all motion vectors without membership, and the above algorithm computes a new suitable initial seed among this set of vectors. The growing-reducing process is applied, and these steps are repeated until all macroblocks have been assigned to a class.

The next sub-section details the seed growing-reducing algorithm and its successive steps.

III.2.1.2 Seed growing-reducing al gorithm

To obtain an efficient segmentation, a new growing region algorithm is presented. The main features of this fully automated algorithm are:

- The macroblock membership can be revised at any time.
- Variance-based motion adaptive thresholds are used to stop the growing process.
- Enlarging and reducing processes are performed to obtain a consistent class distribution.

In the sequel a compact notation is used where the term "macroblock" refers to both the macroblock and the magnitude of the corresponding motion vector. For example, *histogram of the macroblocks* is equivalent to *histogram of the macroblock motion vector magnitudes.*

The seed growing algorithm flow chart is presented in Figure III-8 and its phases are discussed in the following sections.

III.2.1.2.1 Dilation process

As discussed in II.2.2, set-based growing algorithms are usually more efficient at the early stages of the growing process than single-element-based growing algorithms. They analyze efficiently macroblock sets instead of single macroblock when calculating statistic of the class. The statistical results are also more meaningful since they are computed on bigger classes. The proposed algorithm uses a variance-based criterion to append macroblocks to an existing class, thus retaining a set growing process concept.

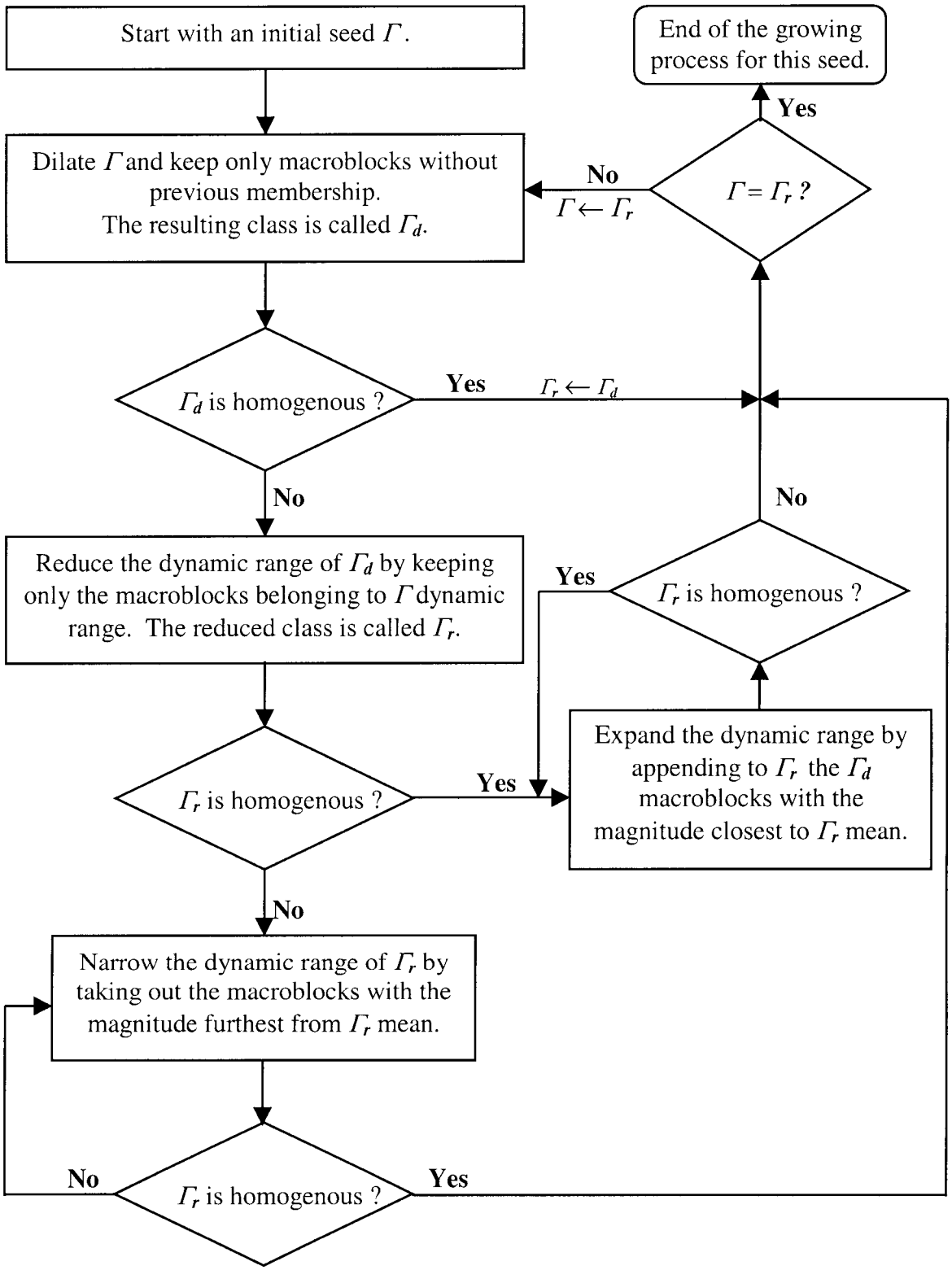


Figure III-8: Seed growing algorithm

As shown in Figure III-9, a 8-Neighboorhood (8N) morphological dilation is performed on the initial seed Γ. A macroblock is said to be in the 8-neighborhood of another one if they are connected along the horizontal, vertical or diagonal direction. Following the dilation, a filtering process extracts only the macroblocks without membership. The resulting set is called Γ_d.

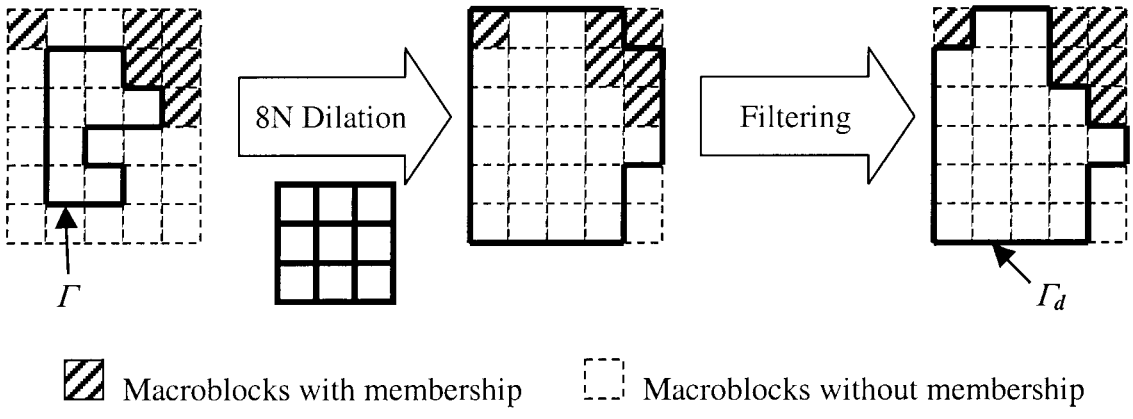


Figure III-9: Example of the dilation-filtering process with a 3x3-neighborhood

III.2.1.2.2 Homogeneous test

A statistical test determines the *homogeneity* of Γ_d. A region Γ_d is said to be homogenous if and only if its motion vector standard deviation $\sigma(\Gamma_d)$ is lower than a motion adaptive threshold, and its motion vector standard deviation variation $\Delta_\sigma(\Gamma_d)$ is lower than a motion-based threshold. More precisely,

$$\Gamma_d \text{ homnogenous} \Leftrightarrow \begin{cases} \sigma(\Gamma_d) \le \sigma_{max}\left(\text{mean}(|\Gamma_d|)\right) \\ \Delta_\sigma(\Gamma_d) \le \Delta_{max}\left(\text{mean}(|\Gamma_d|)\right) \end{cases} \qquad \text{(Eq III-4)}$$
$$\Delta_\sigma(\Gamma_d) = \text{Card}(\Gamma_d)\sigma(\Gamma_d) - \text{Card}(\Gamma')\sigma(\Gamma)$$

The first criterion ensures classes with the concentrate distributions by forcing the class standard deviation to be lower than a motion adaptive threshold. The second criterion

avoids "large class absorbing effects". Consider a large class (composed of many macroblocks) for example the background during the growing process. While appending inconsistent macroblocks to the class, $\sigma(\Gamma_d)$ would not increase significantly and stay below the threshold σ_{max}. The macroblocks would be retained and the class would be composed of non-consistent spread widely motion vectors. The second criterion attenuates this effect by ensuring that the standard deviation variation before and after the dilation stays small. The two threshold functions, σ_{max} and Δ_{max}, use the class mean as parameter. This increases the separation accuracy between moving and non-moving objects by assigning a higher threshold to the class with high mean value. As discussed in III.2, boundary motion vectors in a moving object have usually a lower magnitude than the interior motion vectors. The adaptive threshold combines the object boundary with the object interior and separates it from the background.

III.2.1.2.3 Reduced Dilation Operation

While obtaining a homogenous class, a new dilation is performed, tested for homogeneity and this process is repeated until all macroblocks have been assigned or until the class becomes not homogenous. In the latter case a contraction algorithm is executed. As shown in Figure III-10 in this process all motion vector whose magnitude does not belong to the dynamic range dr(Γ) of Γ are removed. Thus, Γ_d distribution becomes compact while its standard deviation decreases. The reduced class is denoted by Γ_r in the next sub sections.

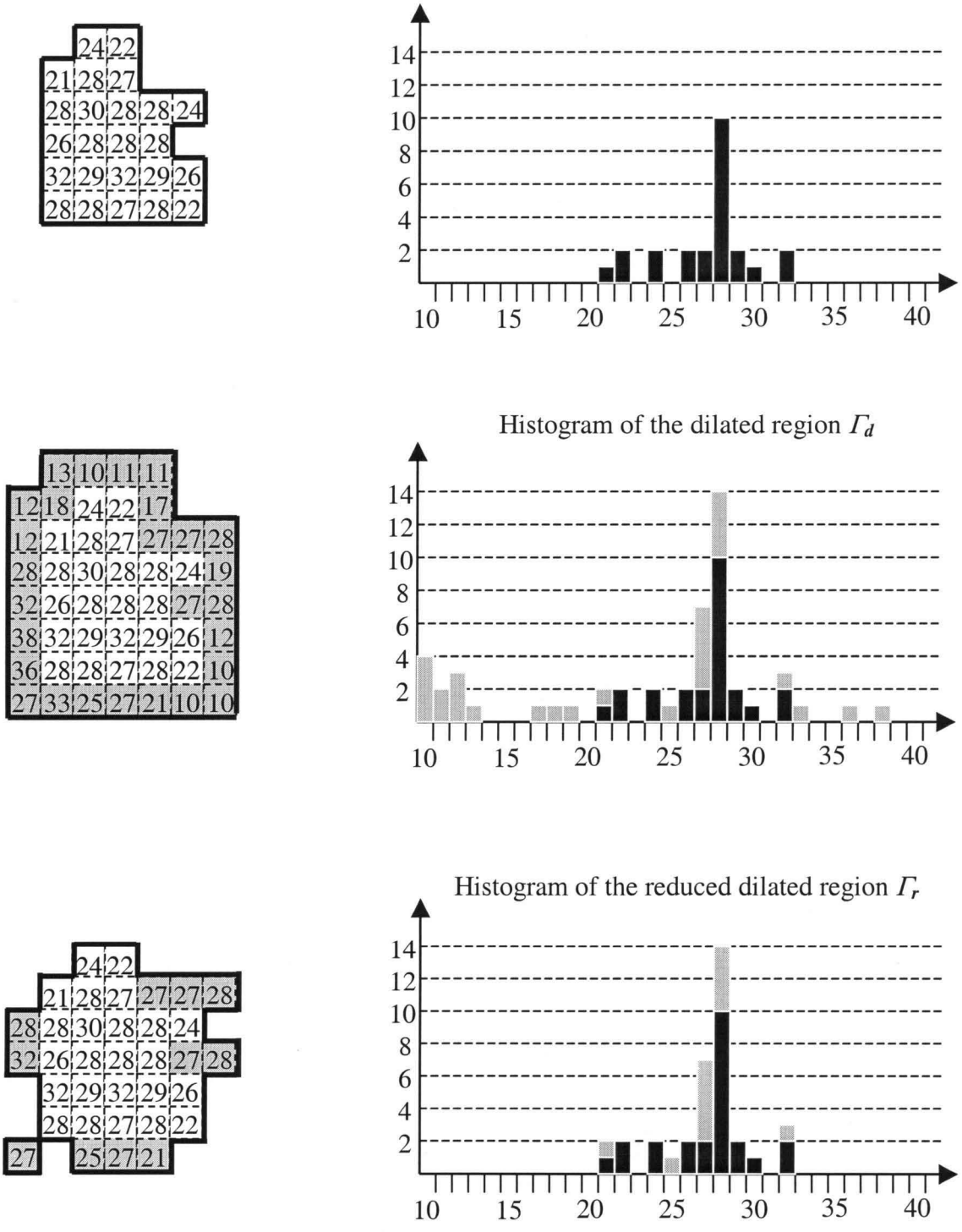


Figure III-10: Dilating-reducing process

III.2.1.2.4 Mean based extension algorithm

The reduced set Γ_r is tested for homogeneity and if the test is positive, the algorithm enlarges the class by increasing its dynamic range. Otherwise, a new contraction step as described in III.2.1.2.5 is performed.

[7] suggests to enlarge the region by appending to Γ_r the neighbor macroblocks contained in $\boldsymbol{L}$. The set $\boldsymbol{L}$ contains macroblocks whose magnitude belongs to the first left non-void bin of dr(Γ). Symmetrically the right extension can be performed using the first right non-void bin of dr(Γ). However this algorithm does not consider the distribution properties of the class Γ_r and may produce inadequate segmentation. Thus the algorithm is modified to use Γ_r distribution information.

To obtain a better segmentation, the proposed extension process computes first the mean $\mu_{\Gamma r}$ of motion vector magnitudes belonging to Γ_r.

$$\mu_{\Gamma_r} = \text{mean}_{m_p(x,y)\in\Gamma_r}\left(V_p(x,y)\right) \qquad \text{(Eq III-5)}$$

The Γ_d macroblocks with the motion vectors closest to $\mu_{\Gamma r}$ are appended to Γ_r. The mean $\mu_{\Gamma r}$ is updated and this process is successively applied to each macroblock of Γ_d until Γ_r becomes inhomogeneous.

Since Γ_d was not homogenous, this will occur inevitably before Γ_r equals Γ_d. The algorithm ensures that the appended macroblocks are consistent with the Γ_r distribution reinforcing meanwhile the compactness of the class. Afterwards, the last homogenous-

enlarged set is compared to the initial set Γ. If these sets are equal, the seed growing process ends, otherwise, it returns to a new dilation process as shown in Figure III-8.

III.2.1.2.5 Mean based contraction algorithm

The reduced dilated class Γ_r obtained in III.2.1.2.3 may not be homogenous. In this case a contraction algorithm is applied. This process follows the rules described in the previous section but removes rather than appends macroblocks.

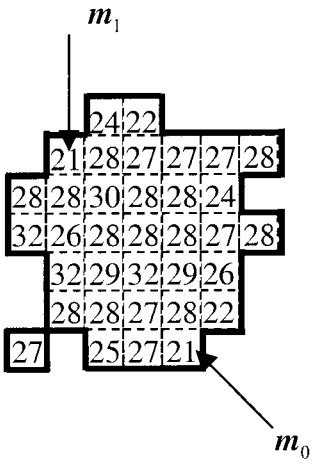
The mean $\mu_{\Gamma r}$ of the class Γ_r is first computed. The Γ_r macroblocks with the motion vectors magnitude furthest to $\mu_{\Gamma r}$ is generated and subtracted from Γ_r. The new contracted class Γ_r is tested for homogeneity. This process is repeated until Γ_r becomes homogenous. Figure III-11 shows an example of this contraction process.

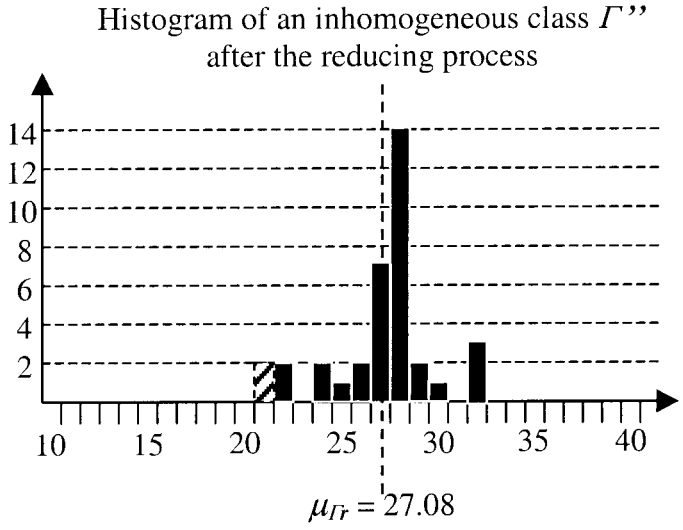
Afterwards, the homogenous contracted set is compared to the initial set Γ. If these two sets are equal, the seed growing process ends, otherwise, it returns to a new dilation process described in the flow chart of Figure III-8.

III.2.1.3 Termination of the spatial growing region algorithm

Following the enlarging or contracting steps, the homogenous class Γ_r is compared to its initial form Γ. If these two sets are identical, the seed growing process stops. A new initial seed is extracted among all macroblocks of Λ_p (macroblocks without membership, see III.2.1.1.) and the growing region algorithm is reapplied.

Finally, when all the macroblocks have been assigned to a region, a basic spatial merging process is performed.

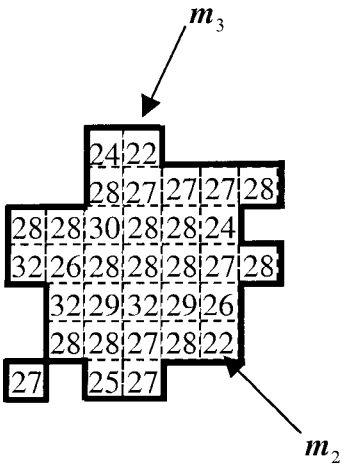


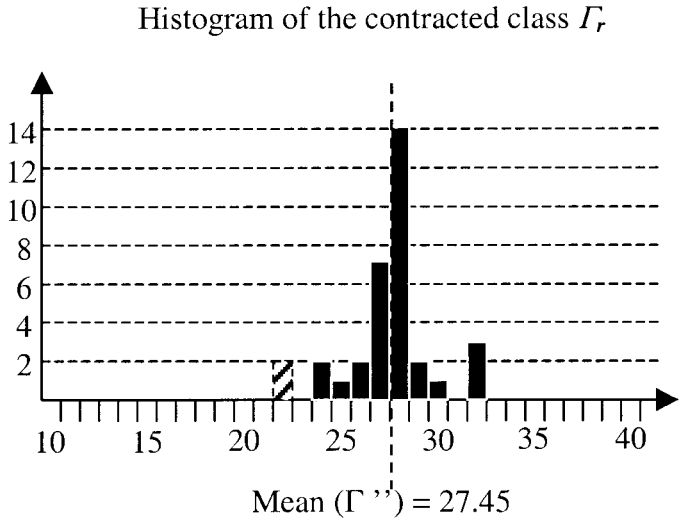


$$\mu_{\Gamma_r} = \text{mean}_{m_p(x,y)\in\Gamma_r}\left(\left\|V_p(x,y)\right\|\right) = 27.08$$

$$\Psi_{\Gamma r} = \left\{m_p(x,y)\in \Gamma_d : \left\|V_p(x,y)\right\| \geq \max_{m_p(x,y)\in\Gamma_d}\left(\left|\left\|V_p(x,y)\right\| - \mu_{\Gamma_r}\right|\right)\right\} = \{m_0, m_1\}$$

Therefore m_0, m_1 are removed from Γ_r.





$$\mu_{\Gamma_r} = \text{mean}_{m_p(x,y)\in\Gamma_r}\left(\left\|V_p(x,y)\right\|\right) = 27.45$$

$$\Psi_{\Gamma r} = \left\{m_p(x,y)\in \Gamma_d : \left\|V_p(x,y)\right\| \geq \max_{m_p(x,y)\in\Gamma_d}\left(\left|\left\|V_p(x,y)\right\| - \mu_{\Gamma_r}\right|\right)\right\} = \{m_3, m_4\}$$

Therefore m_2, m_3 are removed from Γ_r.

Figure III-11: Contraction process example

III.2.2 Small region merging

The spatial classification commonly produces few small regions resulting from the frame border effects or from the noise. These small classes do not affect the spatial segmentation and may be ignored during the summarization process. However, the number of classes increases noticeably and reduced the intra-frame region-merging algorithm efficiency (see III.2.3). Thus a merging process is performed on all small regions to retain only the most significant classes. Since the medium size classes with coherent motion must be retained, only very small classes composed of one or two macroblocks are merged.

Different approaches have been proposed to choose the merging region criteria. Although important, this step is not essential since its only purpose is to increase the intra-frame region matching efficiency. Here, the algorithm stays simple, to minimize processing time. Figure III-12 shows how each small class macroblock is merged with the region corresponding to its *closest* neighboring macroblock. The term *closest* refers to the Euclidean distance between their motion vector magnitudes.

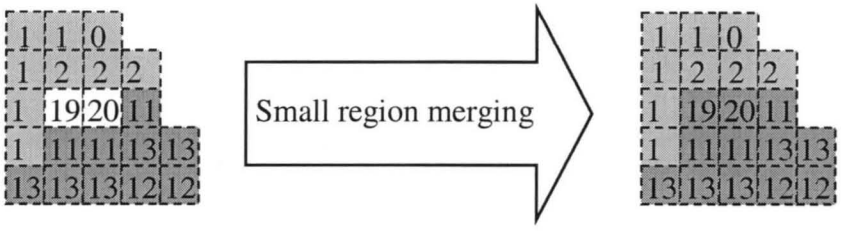


Figure III-12: Region merging example

III.2.3 Intra-Frame region-merging algorithm

III.2.3.1 Algorithm presentation

As shown in Figure III-1, an intra-frame-region-merging algorithm follows the spatial frame segmentation. This process relates regions between frames to obtain a temporal classification. Predictive motion information is used to compute a predicted region position which is compared to actual regions located at this position. A region overlapping metric and a fixed threshold allow to decide whether the two regions represent the same object and thus should be associated. For clarity of presentation the "look-up window" threshold α_L is defined meaning that the search algorithm uses the α_L previous to f_P frames to find a similar region.

As shown in Figure III-13, the proposed algorithm sorts the classes in frames f_i (with $p \leq i \leq p\text{-}\alpha_L$) in order of their decreasing cardinal number (representing the number of macroblocks in each class). This pre-sorting allows to compare first the largest, usually the most relevant, classes and has been shown to improve significantly the region matching accuracy while being computationally efficient.

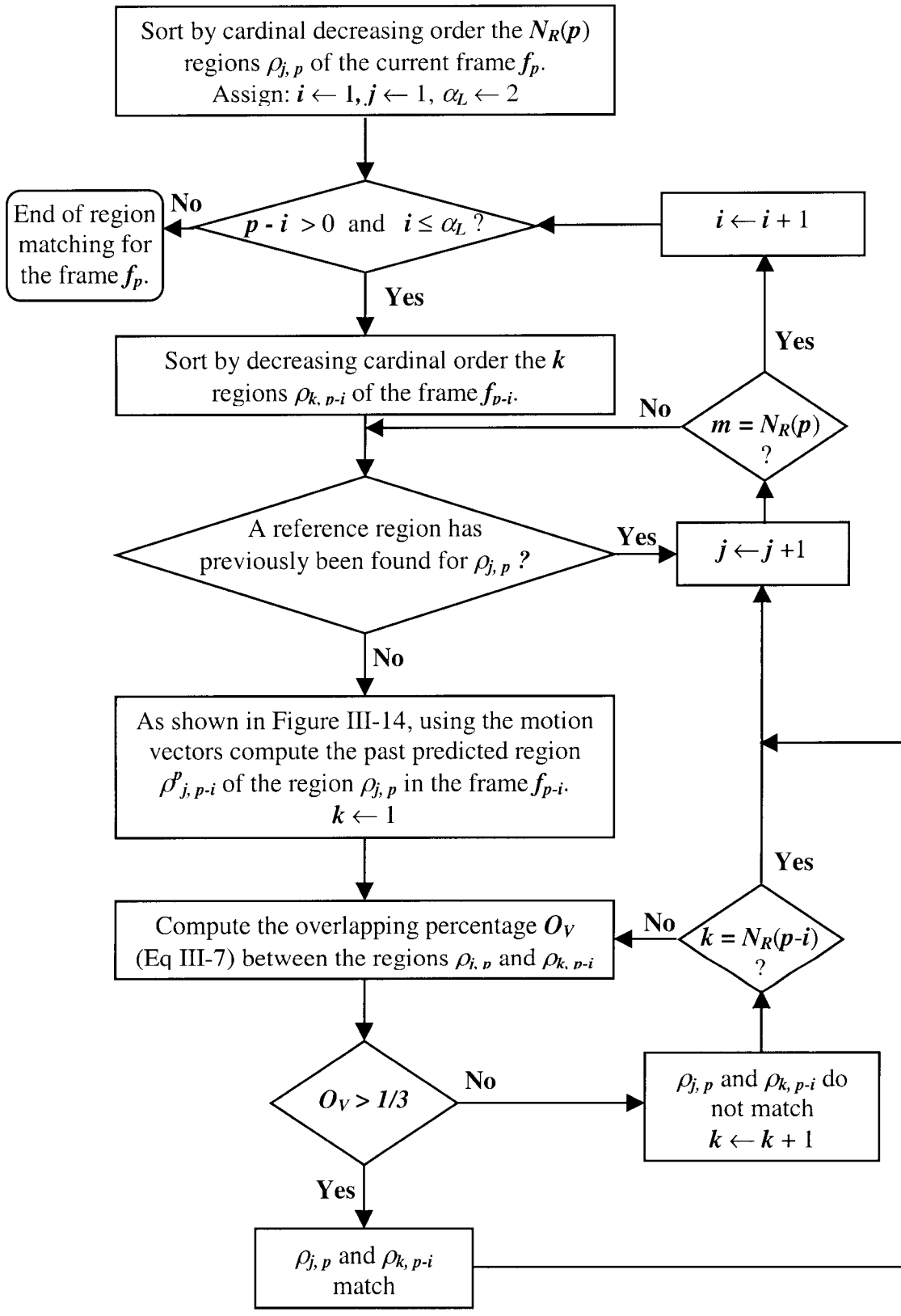
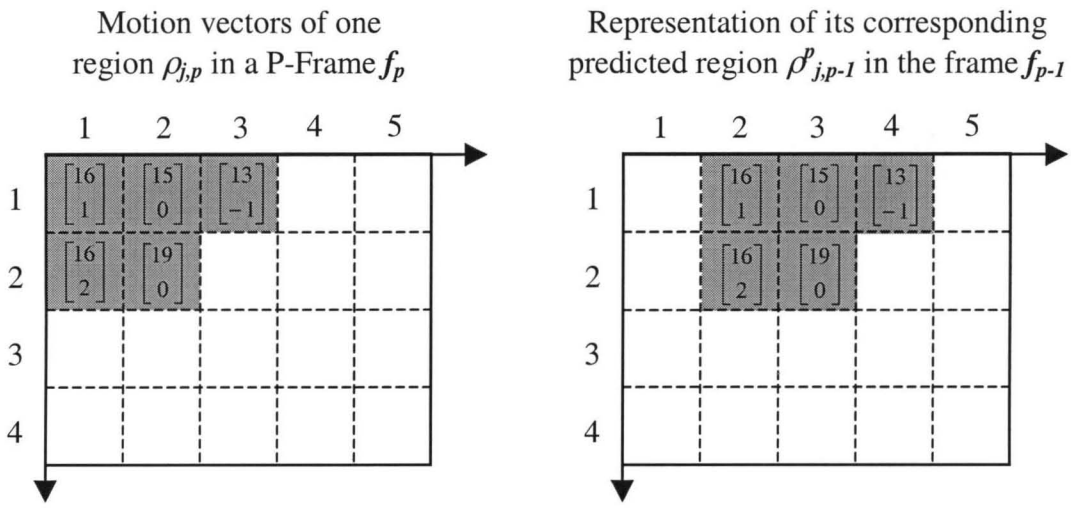


Figure III-13: Time merging algorithm flow chart

The merging algorithm looks up the α_L previous P-frames of f_p and performs an intra-frame region merging. As shown in Figure III-14, for each region $\rho_{j,p}$ of the current frame, the predicted past region $\rho^p_{j,p-i}$ in the frame f_{p-i} (with $1 \le i \le \alpha_L$) is computed using its motion vectors:

$$\rho^p_{j,p-i} = \left\{ m_{p-i}(x,y) \in M_P : \forall m_p(u,v) \in \rho_{j,p}, \left\{ \begin{array}{c} x = u + round\left[\dfrac{i.Vx_p(u,v)}{c}\right] \\ y = v + round\left[\dfrac{i.Vy_p(u,v)}{c}\right] \\ (x,y) \in [1; S_X] \text{x} [1; S_Y] \end{array} \right\} \right\} \quad \text{(Eq III-6)}$$



Example to compute the predicted position $m_{p-1}(u, v)$ of $m_p(1, 1)$:

$$\left\{ \begin{array}{l} u = 1 + \text{round}\left(1\left(\dfrac{16}{16}\right)\right) = 1 + 1 = 2 \\ v = 1 + \text{round}\left(1\left(\dfrac{1}{16}\right)\right) = 1 + 0 = 1 \end{array} \right\}$$

Figure III-14: Predicted region computation example

The motion vector components are scaled by a constant c to provide a motion in macroblock instead of pixel units. The constant value depends on the encoder since it corresponds to the macroblock size measured in pixels. For example, c is set to 16 for encoders using a full-pixel length unit.

Next, a metric O_V provides the minimal spatial overlapping percentage between $\rho^p_{j,p-i}$ and each k region $\rho_{k,p-i}$ of the frame p-i :

$$O_V\left(\rho_{j,p},\rho_{k,p-i}\right)=\frac{\mathrm{Card}\left(\rho^p_{j,p-i}\cap\rho_{k,p-i}\right)}{max\left(\mathrm{card}\left(\rho^p_{j,p-i}\right),\mathrm{card}\left(\rho_{k,p-i}\right)\right)} \qquad \text{(Eq III-7)}$$

A value strictly larger than 1/3 corresponds to a positive match and the regions $\rho_{j,p}$ and $\rho_{k,p-i}$ are merged (i.e., their identification number becomes the same). Otherwise, k is incremented and these comparison are performed for all $\rho_{k,p-i}$ until a positive match is found or all $\rho_{k,p-i}$ have been compared. The threshold value 1/3 has been chosen experimentally. The original value of 1/2 guaranteed the region matching uniqueness, however it was too restrictive. A threshold of 1/3 increases considerably the region merging efficiency without having to solve many merging conflicts. In such cases the first matching is retained, and since classes are pre-sorted, largest classes are first merged.

III.2.3.2 Look-up threshold choice

Due to noise, a reference region may not be detected in the previous frame p-1 however it may be found in more anterior frames. To improve the temporal merging, the "looked-up window" α_L is set to 2, compromising between time efficiency and merging accuracy. The region $\rho^P_{j,p-2}$ is computed by assuming a constant object velocity between the frame p and p-2. In most encoders, two P-frames f_p and f_{p-2} are separated in the worst

case by one I-frame and one P-frame. Considering the NTSC standard (29.97 frames/sec), this interval produces a time interval of 0.3sec. A region merging between these frames indicates that the proposed algorithm assumed a constant object velocity during 0.3sec, missing eventually an object acceleration. However, temporal merging is performed by considering two frames simultaneously and provides a better classification. Choosing a larger α_L value increases the computational time and the time interval discussed above.

III.3 Post segmentation filtering

A classification of the macroblocks is obtained as the result of the previous merging algorithm. Each class φ_j is a macroblock set represented in several frames and defined by:

$$\varphi_j = \bigcup_{p=1}^{p=N} \rho_{j,p} \qquad \text{(Eq III-8)}$$

Noise from background and frame border effects provide an over-segmentation of the video sequence. As an example, a 10s video sequence can be segmented in hundred regions. Summarizing all these regions is useless since many of them are meaningless. Thus, the filter retains only the "main objects" of the video sequence.

Many criteria could be used to define a "main object". In still images, region spatial distribution criteria such as perimeter, surface, standard deviation, etc. allow to identify main objects. In a video sequence, temporal persistence is a confident criterion since noise has no coherent motion. Classes present during many frames usually identify an object with consistent motion. Therefore, the filter retains only classes present more

than one second equivalent to 6 P-frames in the NTSC standard. This duration threshold is experimentally chosen in order to remove noise and identify the footage main objects.

Another approach consists on retaining only the two, or three main objects. Both methods can be implemented. The first one presents the advantage of not being limited to a fixed number of objects while the second one can retain an object appearing during a shorter time than the fixed threshold. Finally a combination of both methods is the most appropriate and left as the future work.

Figure III-15 shows an example of the filtering algorithm applied on the "boat" video sequence (see Table IV-1).

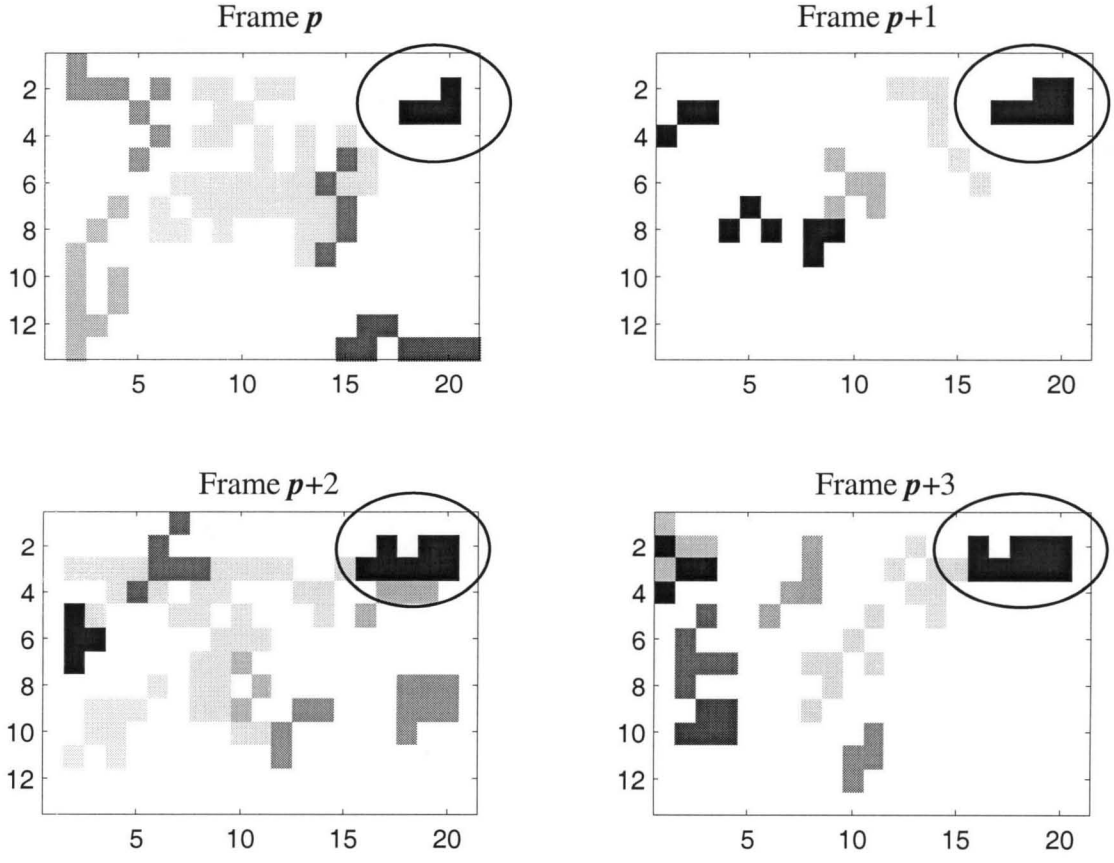


Figure III-15: Segmentation filtering example

Four frames out of the twenty-three have been extracted to demonstrate the spatio-temporal classification. Each gray color corresponds to a different region, and only the white background and the black class on the right top corner keeps exactly the same region number. The other classes differ from a frame to another and after filtering, only the background and the boat remain.

III.4 Summarization

After segmentation and filtering, motion of the remaining objects need to be summarized in the form of model parameters. The model parameters are used to tag the MPEG video file header. Different approaches have been proposed for summarization methods such as text mapping, information histogram, Bayesian models [8], Auto-Regressive Moving Average (ARMA) models, etc.

In this thesis, the information to be summarized is object motion including position, velocity and acceleration. An ARMA model can take advantage of this since its parameters can be related to physical coefficients such as gravity acceleration, friction coefficient, spring elasticity coefficient, etc. This allows the user interface query to have different forms:

- Query by example: the user draws successive object positions, the distance between each position being a velocity indicator. A metrics compares the estimated parameters of both ARMA models: computed from the user input and from the summarization.
- Query using physical coefficients: The user can search for objects which fall under the gravity forces. A metric compares directly the coefficient provided by the user with the ones from the summarization.

The general equation for a ARMA(***n***,***m***) model is the sum of a AR(***n***) model and a MA(***m***) model:

$$y_G[p] = \sum_{i=1}^{n} C_i y_G[p-i] + \sum_{j=1}^{m} D_j w[p-j] \qquad \text{(Eq III-9)}$$

In the proposed summarization, y_G represents the p successive frame coordinates of the region gravity center and w represents white noise.

In this thesis, a ARMA(2,0) model is retained which conveniently summarizes object position, velocity and acceleration:

$$y_G[p] = Ay_G[p-1] + By_G[p-2] \qquad \text{(Eq III-10)}$$

. Matlab estimates the parameters A and B using an iterative Gauss-Newton algorithm which minimizes the prediction error.

The first object to be modeled is the background as it represents the motion reference. Since its position is meaningless, only its average motion is used to compute the ARMA model parameters. To improve the retrieval efficiency, all other objects are modeled using two motion references:

1. The first one uses directly the object gravity center provided by the segmentation. Thus, the object motion is given in the background reference. For example, a user may retrieve all footage consisting of a fixed object in a moving background.
2. The second changes the background to a fixed reference by compensating for the background motion. This model allows to retrieve video sequences by specifying object motion independently of camera motions. The compensated position y_C in the fixed reference is given by:

$$y_C[p] = y_C[p-1] + y_G[p] - y_G[p-1] + V_{BK}[p] \qquad \text{(Eq III-11)}$$

where V_{BK} represents the average motion of the background computed with all macroblocks of the background region.

For each of the series y_C and y_G the estimated parameters of the ARMA model are computed. The MPEG file header tag would consist on the following elements:

- A_{Bk} and B_{Bk} which are estimated ARMA parameters corresponding to the background motion.
- $A_G(i)$ and $B_G(i)$ for each main object i, which correspond to estimated ARMA parameters computed with their gravity center position
- $A_C(i)$ and $B_C(i)$ for each main object i, which correspond to estimated ARMA parameters computed using their compensated gravity center position.

In a typical segmentation where one main object moves against a background, the summarization consists of 6 parameters: two for the background and four for the object.

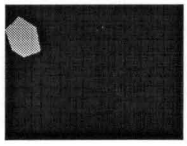
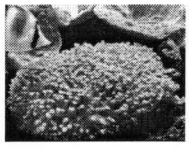
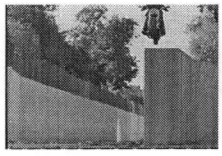
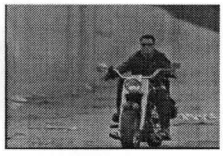
CHAPTER IV : SIMULATION AND RESULTS

IV.1 Video sequence test set

The video sequences test set consists of short sequences only. Long video footages can be easily fragmented into small homogenous video sequences using existing video shot detection methods [22], [23].

All video sequences are MPEG-1 compressed using a 64-pixel-range search algorithm and an IBBPBBPBBPBBI group of picture. The frame rate is chosen to be 29.97 in compliance with the NTSC standard. Finally, the frame sizes as well as the data compression ration differs in all the sequences. Table IV-1 summarizes the sequences used and their different attributes. The first and last frames are shown and a short description explains the type of video sequence motion. More still pictures from these video sequences are shown in Appendix B.

The first two video sequences are artificially created. The first represents color random object moving to the right against a black background. The second shows a similar shape bouncing against the same background. The three remaining sequences are extracted from the real life footages and are sorted in increasing complexity order.

Name	Statistics*	Starting frame	Ending frame	Comments
(1) Colored shape	Dur: 1.47s S: 320x240 DR:172Ko/s			Synthetic footage of a colored shape moving to the right.
(2) Bouncing shape	Dur :2.67s S: 336x208 DR: 153Ko/s			Synthetic shape making one bounce on the bottom.
(3) Boat	Dur :2.67s S: 336x208 DR: 153Ko/s			A boat appears and moves to the left against a relatively fixed background.
(4) Fish	Dur :1.13s S: 352x240 DR: 171Ko/s			A fish crosses the frame diagonal.
(5) Motorbike	Dur :4.93s S: 352x240 DR: 127Ko/s			Down moving background with a zooming effect on a falling motorbike.

*Dur = duration in seconds, S = frame size in pixel, DR = MPEG data rate in Ko/s

Table IV-1: MPEG sequence test set

IV.2 Spatio-temporal segmentation results

The spatial segmentation is one if the main feature of the algorithm. By testing its performance, the following points are verified:

- Input of the segmentation: The motion vector magnitudes provide enough information to perform a suitable segmentation.
- Spatial segmentation: the growing region algorithm separates suitably objects and background.
- Thresholds: while not optimal, they allow an efficient segmentation.

IV.2.1 Spatial segmentation results

As shown in Figure IV-1, the variance based growing region algorithm used to segment frame spatially provides a suitable classification. The "color shape" sequence shows the usefulness of the criteria to avoid "background absorbing effects" as discussed in III.2.1.2.2. Here the background is correctly identified and does not absorb the small object.

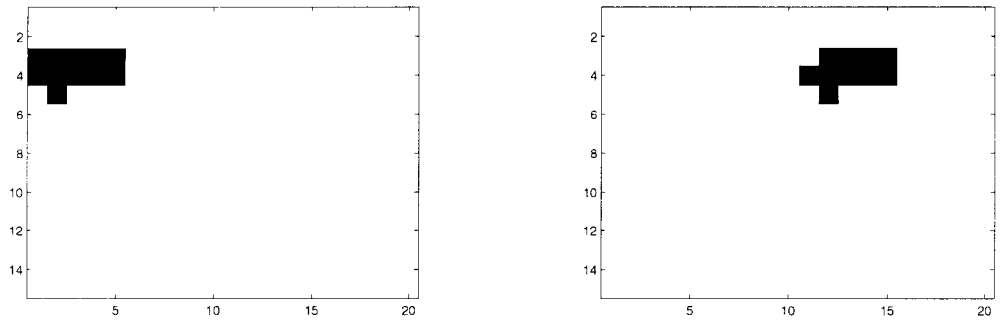


Figure IV-1: Spatial segmentation on a "colored shape" sequence frame

Figure IV-2 shows the spatial classification of the boat sequence. Since the background contains more noise than the previous footage, the resulting classification has more regions. The algorithm suitably identifies the boat, which is the main moving object of the video sequence. However, as discussed in III.2.2, a small region merging process is necessary to obtain a better intra-frame region merging.

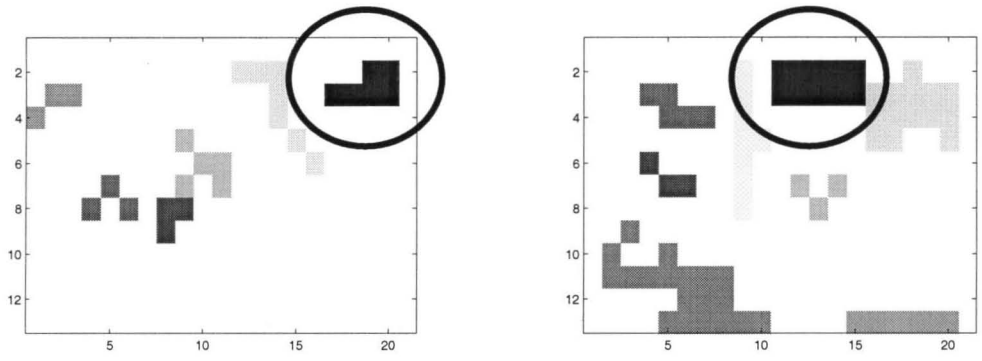


Figure IV-2: Spatial segmentation on a "boat" sequence frame (real footage)

Figure IV-3 shows how the spatial segmentation may become more difficult to accomplish. Although in many frames the algorithm provides a suitable spatial segmentation, the difficulty resides in segmenting the bike and the background when the bike stays stationary during one frame. As shown in Figure IV-4, the bike motion vector magnitudes resemble the background and the spatial segmentation using only this motion becomes inefficient. To obtain a better classification, the motion vector components should be treated separately.

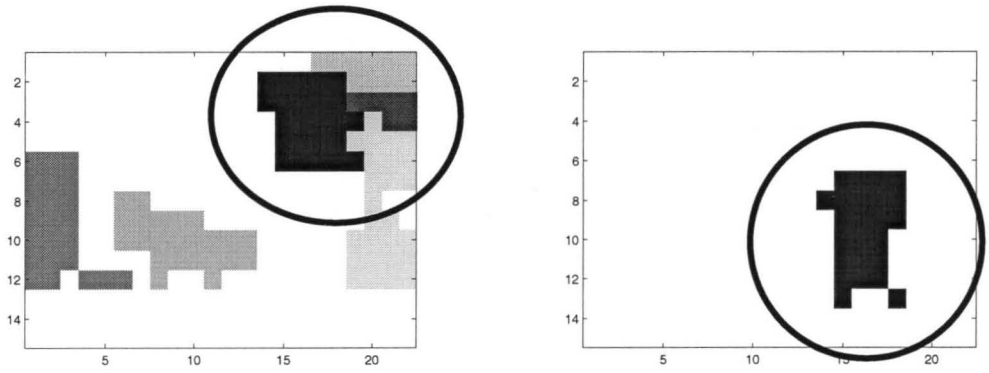


Figure IV-3: Spatial segmentation on a "bike" sequence frame (real footage)

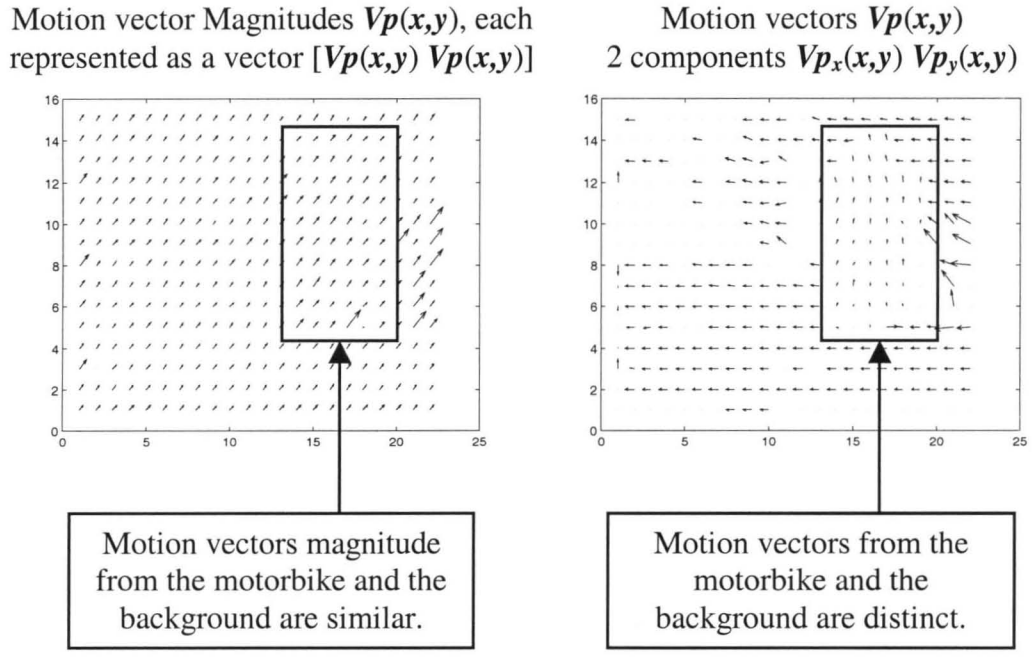


Figure IV-4: Motion vector magnitude versus full motion vector

The previous results show that the motion vectors can be used for retrieval tasks, although it is not their primary purpose. In most cases, the motion vector magnitude in the background reference provide a suitable input for the growing region algorithm. The two homogeneity criterions discussed in III.2.1.2.2 as well as the motion based thresholds allow to obtain compact, homogenous regions.

However, using the motion vector magnitude fails for some specific video sequences. Also the threshold values are not optimal. In few frames, object boundary macroblocks are absorbed by the background region since they have a lower motion vector magnitude. In future work, a more suitable choice of these thresholds could improve the spatial segmentation. In the proposed algorithm, the small regions are merged to improve the time-merging accuracy. To obtain a better classification, the merging algorithm can be improved to merge also larger consistent regions.

IV.2.2 Temporal merging and filtering

The following section discusses the strengths and weaknesses of the temporal merging algorithm. As discussed in III.2.3, the intra-frame region merging algorithm merges classes between frames and identifies object motion attributes along the video sequence. Measuring the merging algorithm performance can be done in many ways. In this section we discuss several successful tests by comparing the region gravity center motion computed by the algorithm and the object motion observed visually.

Figure IV-5 shows the object gravity center motion computed from the two synthetic sequences.

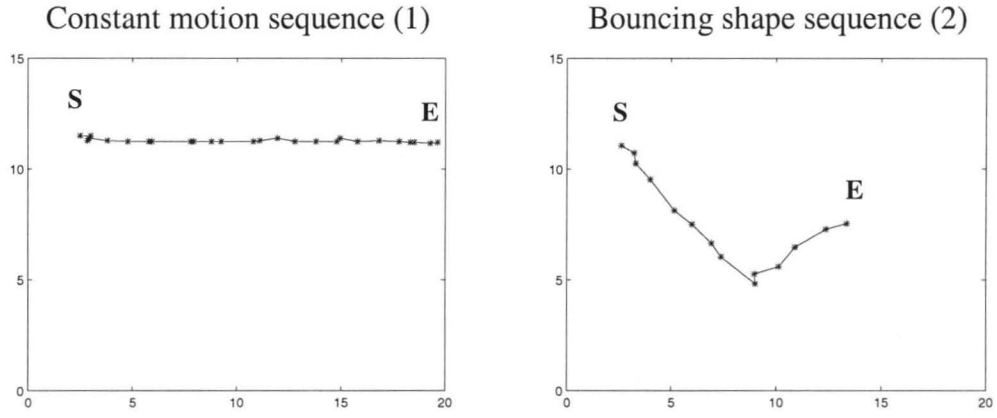


Figure IV-5: Main region gravity center motion on the synthetic sequences

The intra-frame-merging algorithm is efficient allowing to track the main object motion during the whole video sequence. In the second footage, the bouncing (i.e. motion "discontinuity") does not affect the merging although its efficiency decreases during the rebound. Table IV-2 shows the metrics O_V values during the shape region merging. These values decrease significantly during the rebound meaning that in some video sequences, a motion discontinuity can cause region intra-frame merging failures. Thus, a single object could be decomposed in two separate regions.

Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7
-	0.60	0.50	0.60	0.68	0.58	0.63
Frame 8	Frame 9	Frame 10	Frame 11	Frame 12	Frame 13	Frame 14
0.59	0.68	0.37	0.39	0.50	0.66	0.52

Table IV-2: O_V values during the bouncing shape region merging

In the real sequences with smooth, non-accelerated motion, the intra-frame merging provides an efficient segmentation. As shown in Figure IV-6, the algorithm has successfully merged the boat regions of each frame. Since the resulting region last for more than one second, the boat is consistently detected and the algorithm correctly identifies its motion. The same is true for the fish sequence: the fish motion is correctly identified and a retrieval query on these two video sequences would yield a positive match.

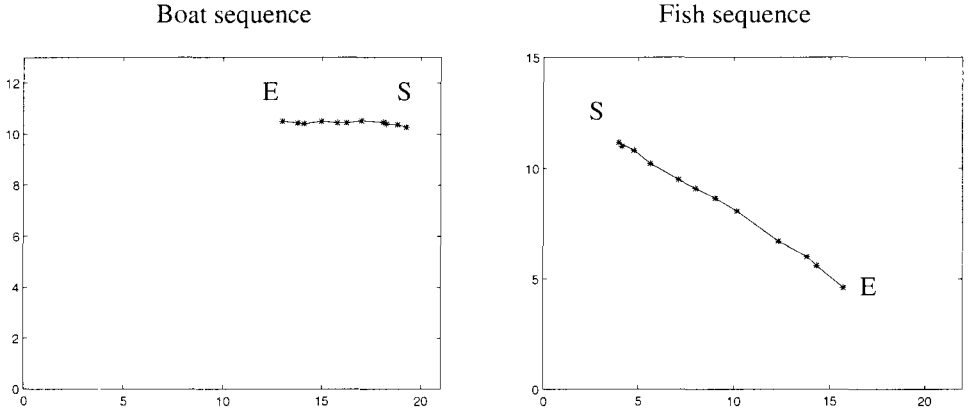


Figure IV-6: Fish and boat gravity centers motion

Figure IV-7 shows the merging-algorithm efficiency in a real sequence which includes down-panning camera effect. The algorithm provides a suitable frame merging that shows the falling bike motion. The bike bouncing is also detected as its gravity center moves up after reaching the ground. However the "second bike fall" is not detected due to a weak spatial segmentation as discussed in the previous section.

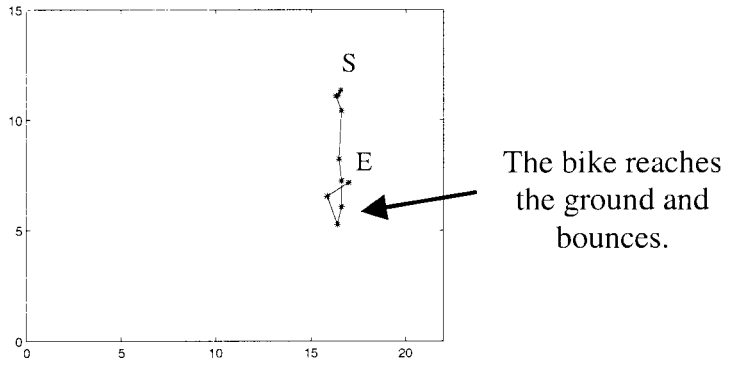


Figure IV-7: Motorbike gravity center motion for the "motorbike" sequence

The intra-frame-merging algorithm provides the final classification. In the real video sequences this merging is important to identify main object motion.

IV.2.3 Summarization

Figure IV-8 shows the difference between the direct and compensated gravity center positions as discussed in IV.2.3. In the "bike" video sequence the camera follows a falling bike creating a panning effect which distorts the true bike motion. By compensating the bike motion with the background average motion, the summarization model coefficients are closer to the real acceleration. However it's important to note that the true laws of physics seldom apply to video sequences. Here, for example, the sequence is played in a slow motion. Other camera effects like zooming would also distort the model coefficient values.

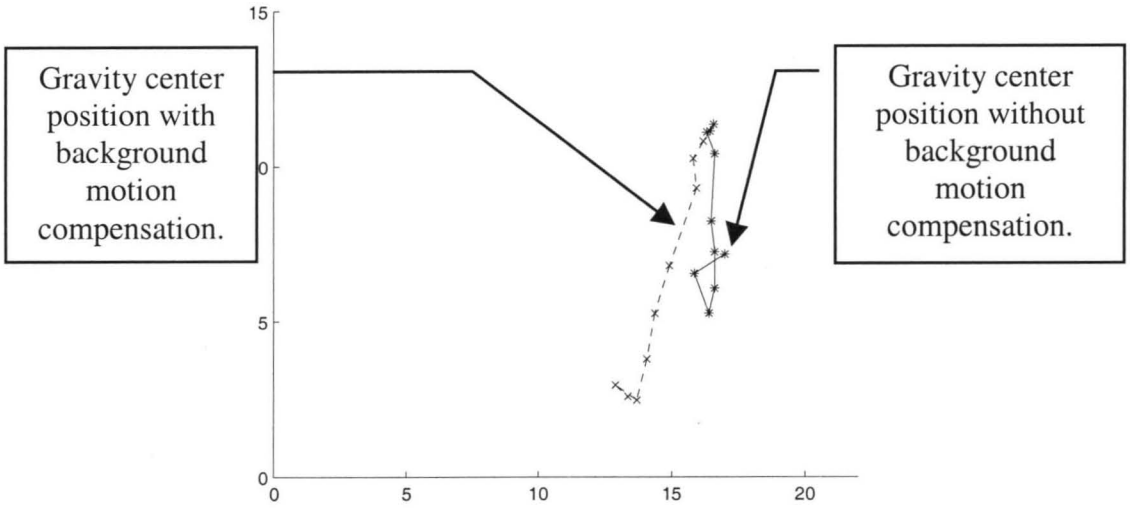


Figure IV-8: Background motion compensation versus no compensation

In this example, the two estimated parameters $A_C(i)$ and $B_C(i)$ of the corresponding ARMA model are :

	$A_C(1)$	$B_C(1)$
Without motion compensation	1.55	0.54
With motion compensation	-0.26	1.26

Table IV-3: ARMA estimated parameters for the "bike" sequence

CHAPTER V : CONCLUSIONS AND RECOMMANDATIONS

V.1 Conclusions

The rapid generation and distribution of digital information and in particular digital video requires development of new technologies for efficient summarization and retrieval. Although many algorithms have been proposed, few are motion based and almost none of them works directly in the MPEG compressed domain.

In this thesis a new fully automated summarization algorithm of MPEG compressed videos has been designed. This algorithm segments the motion vector magnitude from the MPEG stream and provides classes representing the sequence main objects. Initial classes are computed according to the motion vector magnitude histogram. Variance based growing region algorithm expands and contracts the region until reaching the maximum size. After basic spatial merging, time region merging is performed using the predicted motion vectors of each region macroblocks. For each region of the current frame, a backward predicted region is computed and compared to the anterior frame regions. An overlapping metrics indicates whether the region merging must be performed. The result of the segmentation provides a spatio-temporal classification of the motion vectors into different classes, each class corresponding to a different object.

Only time persisting regions are retained to summarize the MPEG video sequence. The summarization consist of four, second order ARMA model parameters computed using the gravity center coordinates in two references: background and fixed. These data should be matched with the user input for the retrieval purposed.

Spatio-temporal segmentation provides correct results. In the tested video sequences, the algorithm filters noise and suitably identifies background and main object motion. The summarization model provides physical motion coefficients although they can be distorted by such video effects as slow motion.

V.2 Recommandations for future work

There are number of areas ripe for continued research in motion based video summarization and retrieval as listed below:

- Spatio-temporal segmentation optimization. Research should focus on motion vector pre-filtering (e.g. removal of frame boundary macroblocks), threshold optimization and intra-frame merging algorithm improvement using for example search areas.
- Compressed color information can be included since MPEG compressed domain provides color attributes. Combining motion vectors with compressed color information should improve the segmentation efficiency and the retrieval accuracy without increasing considerably the algorithm complexity.
- Summarization and retrieval using physical motion models. Research should focus on proper model selection since summarizing motion using such physical concepts as acceleration and friction may yield compact description.

REFERENCES

[1] P. Bouthemy and E. Francois, "Motion Segmentation and Qualitative Dynamic Scene Analysis From an Image Sequence," *International Journal of Computer Vision*, vol. 10, no. 2, pp. 157-182, 1993.

[2] G. Adiv, "Determining Three-Dimensional Motion and Structure From Optical Flow Generated by Several Moving Objects," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 7, no. 4, pp. 384-401, July 1985.

[3] F. Moscheni, S. Bhattacharjee and M. Kunt, "Spatio-temporal Segmentation Based on Region Merging," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 9, Sep 1998.

[4] F. Moscheni and S. Bhattacharje, "Robust Region Merging for Spatio-temporal Segmentation," *IEEE International Conference on Image Processing*, vol. 1, pp. 501-504, Sep 1996.

[5] F. Moscheni and F. Dufaux, "Regions Merging based on robust statistical testing," *Proceedings of SPIE – The International Society for Optical Engineering*, pp. 1118-1129, Mar 1996.

[6] H. Gu, Y. Shirai and M. Asada, "MDL-Based Spatio-temporal Segmentation from Motion in a Long Image Sequence," *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun 1994.

[7] C. Revol and M. Jourlin "New Minimum Variance Region Growing Algorithm for Image Segmentation," *Pattern Recognition Letters*, vol. 18, no 3, pp. 249-258, Mar 1997.

[8] N. Vasconcelos and A. Lippman, "Bayesian modeling of video edition and structure: semantic features for video summairzation and browsing," *IEEE International Conference on Image Processing*, vol. 3, pp. 153-157, Oct 1998.

[9] R. Adams and L. Bischof, "Seeded Region Growing," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 16 no. 6, pp. 641-647, Jun 1994.

[10] S. Chang, J. Smith, M. Beigi and A. Benitez "Visual information retrieval from large distributed online repositories," *Communications of the ACM*, vol. 40, no. 12, pp. 63-71, Dec 1997.

[11] M. Swain, 'Interactive Indexing into Image Databases," *Proceedings of SPIE – The International Society for Optical Engineering*, vol. 1908, pp. 95-103, Feb 1993.

[12] J. Krey, M. Roper, P. Alshult, Th. Hermes and O. Herzog, "Video retrieval by still image analysis with ImageMiner^{TM}," *Computer & Graphics (Pergamon)*, vol. 22, no. 6, pp. 675-685, Dec 1998.

[13] N. Dimitrova and M. Abdel-Mottaleb, "Content-based video retrieval by example video clip," *Proceedings of SPIE – The International Society for Optical Engineering*, vol. 3022, pp. 59-70.

[14] Y. Deng and B.S. Manjunath, "Content-based search of video using color, texture, and motion," *IEEE International Conference on Image Processing*, vol. 2, pp. 534-537, Oct 1997.

[15] S. Chang and J. Smith, "Visually searching the Web for content," *IEEE Multimedia,* vol. 4, no. 3, pp. 12-20, Jul-Sep 1997.

[16] A. Tremeau and N. Borel, "Region growing and merging algorithm to color segmentation," *Pattern Recognition,* vol. 30, no. 7, pp. 1191-1203, Jul 1997.

[17] G. Fornaro and D. Rossi, "Region growing strategy applied to least square phase unwrapping," *Proceedings of IEEE International Geoscience and Remote Sensing Symposium,* Part 1 (of 5), vol. 1, no. 7, pp. 68-70, Jul 1998.

[18] S.A. Hojjatoleslami and D. Rossi," Region growing A new approach," *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1079-1084, Jul 1998.

[19] A. Mehnert and P. Jackway, "Improved seeded region growing algorithm," *Pattern Recognition Letters*, vol. 18, no. 10, pp. 1065-1071, Oct 1997.

[20] Y.L. Chang and X. Li, "Adaptive image region-growing," *IEEE Transactions of Image Processing*, pp. 1057-7149, Nov 1994.

[21] T. Asano and N. Yokoya, "Image segmentation schema for low-level computer vision", *Pattern Recognition*, vol. 14, no. 1-6, pp. 267-273, 1980

[22] T. Kuo, Y.B. Lin, Arbee L.P. Chen, S.C. Chen and C.Y. Ni "Efficient shot change detection on compressed video data," *Proceedings of the International Workshop on Multi-Media Database Management Systems*, pp. 101-108, Aug 1996.

[23] B.C. Song and J.B. Ra, "Fast edge map extraction from MPEG compressed video data for video parsing," *Proceedings of SPIE – The International Society for Optical Engineering*, vol. 3656, pp. 710-721, Jan 1999.

[24] V. Roberto "Intelligent perceptual systems: new directions in computational perception," *Berlin ; New York : springler-Verlag*, 1993.

[25] L. Ljung "System identification, theory for the user," *Englewood Cliffs, NJ : Prentice-Hall*, 1987.

[26] Tektronix "A guide to MPEG fundamentals and protocols analysis (including DVB and ATSC)," *White paper - Tektronix*, 1997.

APPENDICES

APPENDIX A: Computation of the ARMA model estimated parameters

This appendix provides an overview of the Matlab method used to estimate the parameters A and B of the ARMA model:

$$y_G[p] = Ay_G[p-1] + By_G[p-2] + w[p]$$

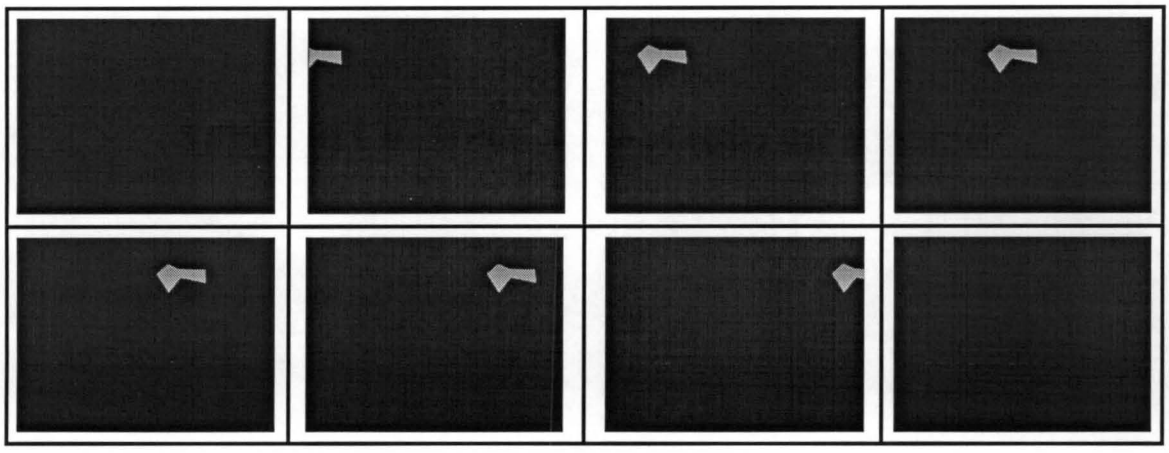
The coefficients of the polynomials are estimated using a prediction error/maximum likelihood method [25], by minimizing the size of the error term w in the above expression. A robust quadratic prediction error criterion is minimized using an iterative Gauss-Newton algorithm. The Gauss-Newton vector is bisected up to 10 times until a lower value of the criterion is found. If no such value is found, a gradient search direction is used instead, and the procedure is repeated. The iterations are terminated when one of these three conditions is true:

- The maximum number of iterations performed during a search for a minimum is reached (the default value is 10).
- The Gauss-Newton vector has a norm less than a threshold.
- A lower value of the criterion cannot be found.

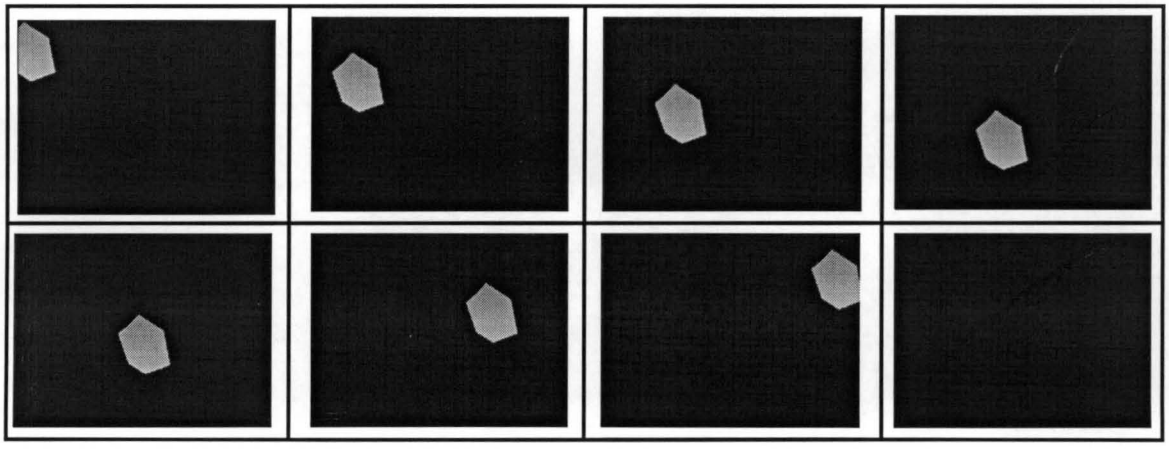
If not specified, the initial conditions for the iterative search are constructed in a special four-stage LS-IV algorithm (see [25] equation 10.75). A stability test of the predictor is performed to assure that only models corresponding to stable predictors are tested.

APPENDIX B: MPEG video samples use as test set

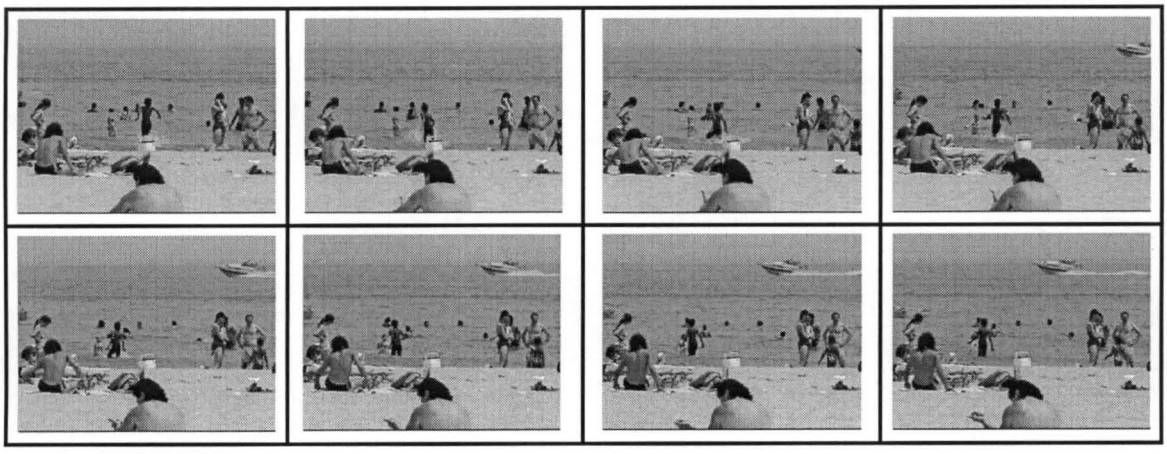
Video Sequence 1: "Colored shape"



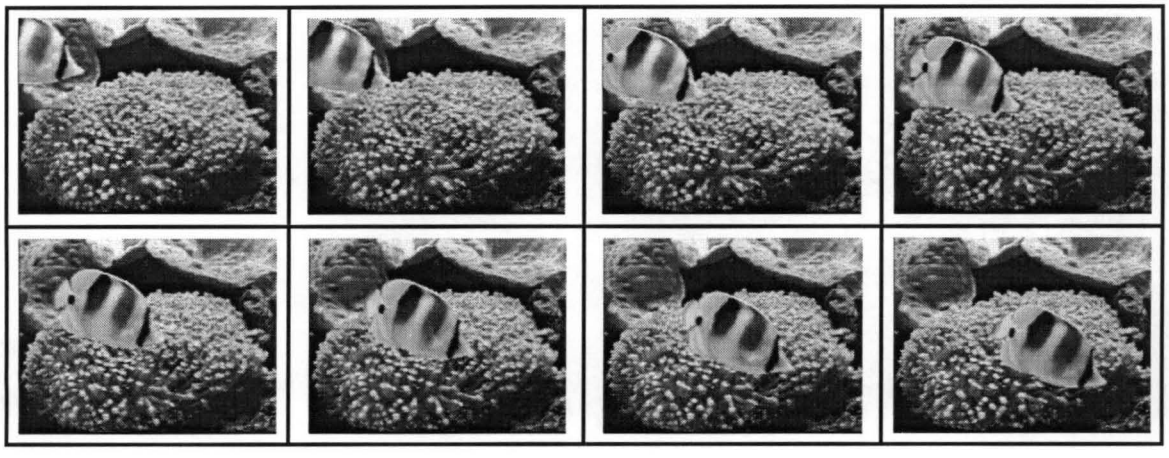
Video Sequence 2: "Bouncing colored shape"



Video Sequence 3: "Boat"



Video Sequence 3: "Fish"



Video Sequence 5: Falling bike